

02-03-00

2723

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of **AKIO OHBA**

U.S. Serial No.: **09/315,713**

Filed : **MAY 20, 1999**

Title : **IMAGE PROCESSING DEVICE, METHOD AND DISTRIBUTION
MEDIUM**

Assistant Commissioner for Patents
Washington, D.C. 20231

SUBMISSION OF PRIORITY DOCUMENT

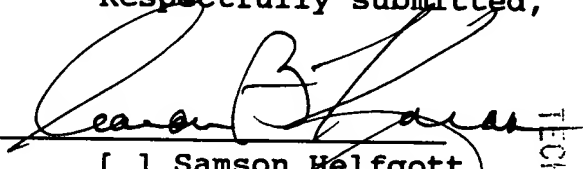
S I R:

Attached herewith is JAPANESE patent application
no. 10-138043 filed MAY 20, 1998 of whose priority has been
claimed in the present application.

Any fee, due as a result of this paper, not covered by
an enclosed check, may be charged to Deposit Acct. No. 08-
1634.

Respectfully submitted,

Any fee due with this paper, not fully
covered by an enclosed check, may be
charged on Deposit Acct. No. 08-1634


[] Samson Helfgott
Reg. No. 23,072
[X] Aaron B. Karas
Reg. No. 18,923

TECH CENTER 2700

RECEIVED
FEB - 7 2000

HELFGOTT & KARAS, P.C.
60TH FLOOR
EMPIRE STATE BUILDING
NEW YORK, NEW YORK 10118
DATE: FEBRUARY 2, 2000
DOCKET NO.: SCEI 16.084
(212) 643-5000

Filed by Express Mail
(Receipt No. E152238687105)
on 2/2/00
pursuant to 37 C.F.R. 1.10.
by Angela Haussermann



日本国特許庁
PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されて
いる事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed
with this Office.

出願年月日

Date of Application:

1998年 5月20日

出願番号

Application Number:

平成10年特許願第138043号

願 人

Applicant(s):

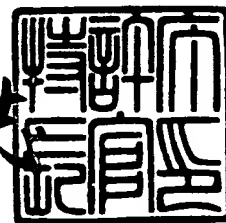
株式会社ソニー・コンピュータエンタテインメント

RECEIVED
FEB-7 2000
TECH CENTER 2700

1999年 6月11日

特許庁長官
Commissioner,
Patent Office

伴佐山建志



出証番号 出証特平11-3038959

【書類名】 特許願

【整理番号】 9800435909

【提出日】 平成10年 5月20日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 19/00

【発明の名称】 画像処理装置および方法、並びに提供媒体

【請求項の数】 27

【発明者】

 【住所又は居所】 東京都港区赤坂7丁目1番1号 株式会社ソニー・コンピュータエンタテインメント内

 【氏名】 大場 章男

【特許出願人】

 【識別番号】 395015319

 【氏名又は名称】 株式会社ソニー・コンピュータエンタテインメント

 【代表者】 徳中 暉久

【代理人】

 【識別番号】 100082131

 【弁理士】

 【氏名又は名称】 稲本 義雄

 【電話番号】 03-3369-6479

【手数料の表示】

 【予納台帳番号】 032089

 【納付金額】 21,000円

【提出物件の目録】

 【物件名】 明細書 1

 【物件名】 図面 1

 【物件名】 要約書 1

 【包括委任状番号】 9504854

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 画像処理装置および方法、並びに提供媒体

【特許請求の範囲】

【請求項 1】 ソースの画像データを画素単位で記憶する第 1 の記憶手段と

、デスティネーションの画像データを画素単位で記憶する第 2 の記憶手段と、

前記第 1 の記憶手段に記憶されたソースの画像データに、画素単位で所定の演算を施して前記第 2 の記憶手段にデスティネーションの画像データとしてポリゴン単位で描画する動作を、所定の演算結果が得られるまで繰り返し行う描画手段と

を備えることを特徴とする画像処理装置。

【請求項 2】 前記第 1 の記憶手段が記憶するソースの画像データは、ビデオカメラが出力する画像データである

ことを特徴とする請求項 1 に記載の画像処理装置。

【請求項 3】 前記ソースの画像データと、前記デスティネーションの画像データとの間の演算モードを指定する指定手段を

さらに備えることを特徴とする請求項 1 に記載の画像処理装置。

【請求項 4】 前記指定手段は、前記演算モードとして、前記ソースの画像データを、前記デスティネーションの画像データに加算する第 1 のモード、または、前記ソースの画像データを、前記デスティネーションの画像データから減算する第 2 のモードを指定する

ことを特徴とする請求項 3 に記載の画像処理装置。

【請求項 5】 前記指定手段は、前記演算モードとして、前記ソースの画像データを、前記デスティネーションの画像データとして、前記第 2 の記憶手段に記憶させる第 3 のモードをさらに指定する

ことを特徴とする請求項 4 に記載の画像処理装置。

【請求項 6】 前記演算は、コンボルーションフィルタ処理、ピラミッドフィルタ処理、フレーム間差分処理、画像間距離演算処理、ハフ変換処理、モーションブラー処理、またはバイリニア補間処理のいずれかである

ことを特徴とする請求項 1 に記載の画像処理装置。

【請求項 7】 前記画像処理装置は、コンピュータエンタテインメント装置である

ことを特徴とする請求項 1 に記載の画像処理装置。

【請求項 8】 ソースの画像の画像データを画素単位で記憶する第 1 の記憶手段と、

デスティネーションの画像データを画素単位で記憶する第 2 の記憶手段とを備える画像処理装置の画像処理方法において、

前記第 1 の記憶手段に記憶されたソースの画像データに、画素単位で所定の演算を施して前記第 2 の記憶手段にデスティネーションの画像データとしてポリゴン単位で描画する動作を、所定の演算結果が得られるまで繰り返し行う描画ステップ

を含むことを特徴とする画像処理方法。

【請求項 9】 ソースの画像の画像データを画素単位で記憶する第 1 の記憶手段と、

デスティネーションの画像データを画素単位で記憶する第 2 の記憶手段とを備える画像処理装置に、

前記第 1 の記憶手段に記憶されたソースの画像データに、画素単位で所定の演算を施して前記第 2 の記憶手段にデスティネーションの画像データとしてポリゴン単位で描画する動作を、所定の演算結果が得られるまで繰り返し行う描画ステップ

を含む処理を実行させるプログラムを提供することを特徴とする提供媒体。

【請求項 10】 ソースの画像データを画素単位で記憶する第 1 の記憶部と、デスティネーションの画像データを画素単位で記憶する第 2 の記憶部とを備える記憶手段と、

前記第 1 の記憶部に記憶されたソースの画像データに、画素単位で所定の演算を施して前記第 2 の記憶部にデスティネーションの画像データとしてポリゴン単位で描画する動作を、所定の演算結果が得られるまで繰り返し行わせる描画コマンドを生成する生成手段と、

前記生成手段により生成された描画コマンドを実行する実行手段と
を備えることを特徴とする画像処理装置。

【請求項 11】 前記第 1 の記憶部に記憶されるソースの画像データは、ビデオカメラが出力する画像データである
ことを特徴とする請求項 10 に記載の画像処理装置。

【請求項 12】 前記生成手段は、前記ソースの画像データと、前記デスティネーションの画像データとの間の演算モードをさらに指定する
ことを特徴とする請求項 10 に記載の画像処理装置。

【請求項 13】 前記生成手段は、前記演算モードとして、前記ソースの画像データを、前記デスティネーションの画像データに加算する第 1 のモード、または、前記ソースの画像データを、前記デスティネーションの画像データから減算する第 2 のモードを指定する
ことを特徴とする請求項 12 に記載の画像処理装置。

【請求項 14】 前記生成手段は、前記演算モードとして、ソースの画像データを、前記デスティネーションの画像データとして前記第 2 の記憶部に記憶させる第 3 のモードをさらに指定する
ことを特徴とする請求項 13 に記載の画像処理装置。

【請求項 15】 前記演算は、コンボルーションフィルタ処理、ピラミッドフィルタ処理、フレーム間差分処理、画像間距離演算処理、ハフ変換処理、モーションブラー処理、またはバイリニア補間処理のいずれかである
ことを特徴とする請求項 10 に記載の画像処理装置。

【請求項 16】 前記画像処理装置は、コンピュータエンタテインメント装置である
ことを特徴とする請求項 10 に記載の画像処理装置。

【請求項 17】 画像データを記憶する記憶部を有する画像処理装置の画像処理方法において、

ソースの画像データを画素単位で第 1 の記憶部に記憶させるとともに、デスティネーションの画像データを画素単位で第 2 の記憶部に記憶させる記憶ステップと、

前記記憶ステップで、前記第1の記憶部に記憶されたソースの画像データに、画素単位で所定の演算を施して前記第2の記憶部にデスティネーションの画像データとしてポリゴン単位で描画する動作を、所定の演算結果が得られるまで繰り返し行わせる描画コマンドを生成する生成ステップと

を含むことを特徴とする画像処理方法。

【請求項18】 画像データを記憶する記憶部を有する画像処理装置に、ソースの画像データを画素単位で第1の記憶部に記憶させるとともに、デスティネーションの画像データを画素単位で第2の記憶部に記憶させる記憶ステップと、

前記記憶ステップで、前記第1の記憶部に記憶されたソースの画像データに、画素単位で所定の演算を施して前記第2の記憶部にデスティネーションの画像データとしてポリゴン単位で描画する動作を、所定の演算結果が得られるまで繰り返し行わせる描画コマンドを生成する生成ステップと

を含む処理を実行させるプログラムを提供することを特徴とする提供媒体。

【請求項19】 ソースの画像データを画素単位で記憶する第1の記憶手段と、

デスティネーションの画像データを画素単位で記憶する第2の記憶手段と、

前記第1の記憶手段に記憶されたソースの画像データに、画素単位で所定の演算のうちの一部の演算を施して前記第2の記憶手段にデスティネーションの画像データとしてポリゴン単位で描画する第1の描画手段と、

前記第1の記憶手段に記憶されたソースの画像データに、画素単位で所定の演算のうちの一部の演算を施して、前記第1の描画手段により既に描画されている画像データに加算または減算して、前記第2の記憶手段にデスティネーションの画像データとしてポリゴン単位で描画する第2の描画手段と

を備えることを特徴とする画像処理装置。

【請求項20】 前記第1の記憶手段記憶されるソースの画像データは、ビデオカメラが出力する画像データである

ことを特徴とする請求項19に記載の画像処理装置。

【請求項 21】 前記ソースの画像データと、前記デスティネーションの画像データとの間の演算モードを指定する指定手段を

さらに備えることを特徴とする請求項 19 に記載の画像処理装置。

【請求項 22】 前記指定手段は、前記演算モードとして、前記ソースの画像データを、前記デスティネーションの画像データに加算する第 1 のモード、または、前記ソースの画像データを、前記デスティネーションの画像データから減算する第 2 のモードを指定する

ことを特徴とする請求項 21 に記載の画像処理装置。

【請求項 23】 前記指定手段は、前記演算モードとして、前記ソースの画像データを、前記デスティネーションの画像データとして、前記第 2 の記憶手段に記憶させる第 3 のモードをさらに指定する

ことを特徴とする請求項 22 に記載の画像処理装置。

【請求項 24】 前記演算は、コンボルーションフィルタ処理、ピラミッドフィルタ処理、フレーム間差分処理、画像間距離演算処理、ハフ変換処理、モーションブラー処理、またはバイリニア補間処理のいずれかである

ことを特徴とする請求項 19 に記載の画像処理装置。

【請求項 25】 前記画像処理装置は、コンピュータエンタテインメント装置である

ことを特徴とする請求項 19 に記載の画像処理装置。

【請求項 26】 ソースの画像データを画素単位で記憶する第 1 の記憶手段と、

デスティネーションの画像データを画素単位で記憶する第 2 の記憶手段とを備える画像処理装置の画像処理方法において、

前記第 1 の記憶手段に記憶されたソースの画像データに、画素単位で所定の演算のうちの一部の演算を施して前記第 2 の記憶手段にデスティネーションの画像データとしてポリゴン単位で描画する第 1 の描画ステップと、

前記第 1 の記憶手段に記憶されたソースの画像データに、画素単位で所定の演算のうちの一部の演算を施して、前記第 1 の描画ステップで既に描画されている画像データに加算または減算して、前記第 2 の記憶手段にデスティネーショ

ンの画像データとしてポリゴン単位で描画する第2の描画ステップと
を含むことを特徴とする画像処理方法。

【請求項27】 ソースの画像データを画素単位で記憶する第1の記憶手段と、

デスティネーションの画像データを画素単位で記憶する第2の記憶手段と
を備える画像処理装置に、

前記第1の記憶手段に記憶されたソースの画像データに、画素単位で所定の演算のうちの一部の演算を施して前記第1の記憶手段にデスティネーションの画像データとしてポリゴン単位で描画する第1の描画ステップと、

前記第1の記憶手段に記憶されたソースの画像データに、画素単位で所定の演算のうちの一部の演算を施して、前記第1の描画ステップで既に描画されている画像データに加算または減算して、前記第2の記憶手段にデスティネーションの画像データとしてポリゴン単位で描画する第2の描画ステップと

を含む処理を実行させるプログラムを提供することを特徴とする提供媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、画像処理装置および方法、並びに提供媒体に関し、特に、簡単な構成で、低コストの装置で、画像処理を行うことができるようにした画像処理装置および方法、並びに提供媒体に関する。

【0002】

【従来の技術】

図46は、従来の画像処理装置の構成例を表している。ビデオカメラ1は、図示せぬ被写体を撮像し、その被写体の画像データをイメージ処理部2とCPU5に出力している。イメージ処理部2は、ビデオカメラ1より入力された画像データをメモリ11に記憶する。CPU5が、イメージ処理部2の演算部12に対して、所定の演算を指令すると、演算部12は、この指令に対応して、メモリ11に記憶されている画像データに対して所定の演算を施し、演算後のデータをグラフィックス処理部3に出力する。

【0003】

グラフィックス処理部3は、イメージ処理部2より入力された演算データをメモリ21に記憶する。CPU5は、グラフィックス処理部3の表示データ生成部22を制御し、メモリ21に記憶されている演算データから表示データを生成させる。グラフィックス処理部3で生成された表示データは、CRT4に出力され、表示される。

【0004】

CPU5はまた、ビデオカメラ1またはイメージ処理部2より出力されたデータの供給を受け、必要に応じて所定の処理を施し、イメージ処理部2またはグラフィックス処理部3に出力する。

【0005】

このような画像処理装置において、例えば、ビデオカメラ1より出力された画像データに対して、コンボルーションフィルタ処理を施し、CRT4に出力し、表示させる処理を行うものとする、そのときの動作は、例えば図47のフローチャートに示すようになる。なお、この処理は、基本的にCPU5により実行される。

【0006】

CPU5は、ステップS1において、ビデオカメラ1より1フレーム分の画像データの供給を受ける。この画像データはソースフレームの画像データとされ、例えば図48に示すように、 $HMAX \times VMAX$ 個の画素データ C_{sp} で構成されている。ステップS1において、CPU5はまた、コンボルーションフィルタ係数 $C_v[m][n]$ を設定する。このコンボルーションフィルタ係数 $C_v[m][n]$ は、図49の例の場合、 3×3 個の係数で構成されている。

【0007】

次に、ステップS2において、CPU5は、デスティネーションフレームの画素データ C_{dp} として、 $HMAX \times VMAX$ 個の画素データを図50に示すように想定し、そのうちの座標(1, 1)の画素データ $C_{dp}[1][1]$ の値を0に初期設定する。さらに、ステップS3において、変数jに1を初期設定し、ステップS4において、変数iに1を初期設定する。この変数i, jは、図48と図50に示すよ

うに、ソースフレームの画像（図48）およびデスティネーションフレームの画像（図50）の水平方向の座標（ i ）または垂直方向の座標（ j ）を表している。 i は、0からHMAX-1までの値を取り、 j は、0からVMAX-1までの値を取る。

【0008】

ステップS5とステップS6においては、変数 n と変数 m に、それぞれ0が初期設定される。この変数 m と n は、図49に示すように、コンボルーションフィルタ係数の水平方向の座標（ m ）と垂直方向の座標（ n ）を表している。この例の場合、 m と n は、いずれも0から2までの値を取る。

【0009】

次に、ステップS7において、CPU5は、次式に示す演算を実行する。

$$C_{dp}[i][j] = C_{dp}[i][j] + C_v[m][n] * C_{sp}[i+m-1][j+n-1] \quad (1)$$

【0010】

いま、 $j=1$ 、 $i=1$ 、 $n=0$ 、 $m=0$ とされ、 $C_{dp}[i][j]$ の初期値は0とされているから、次式が得られる。

$$C_{dp}[1][1] = C_v[0][0] * C_{sp}[0][0] \quad (2)$$

【0011】

次に、ステップS8に進み、CPU5は、変数 m が2より小さいか否かを判定する。いまの場合、 $m=0$ であり、2より小さいので、ステップS9に進み、 m は1だけインクリメントされ、 $m=1$ とされる。そして、ステップS7に戻り、再び上記（1）式の演算が行われる。その結果、次式が得られる。

$$C_{dp}[1][1] = C_{dp}[1][1] + C_v[1][0] * C_{sp}[1][0] \quad (3)$$

【0012】

なお、ここで、上記（3）式の右辺の $C_{dp}[1][1]$ の値は、上記（2）式で得られた値である。

【0013】

そして、ステップS8に進み、再び変数mが2より小さいか否かが判定される。いまの場合、 $m=1$ であるから、ステップS9に進み、変数mがさらに1だけインクリメントされ、 $m=2$ とされる。その後、ステップS7に戻り、再び上記(1)式が演算され、次式が得られる。

$$C_{dp}[1][1] = C_{dp}[1][1] + C_v[2][0] * C_{sp}[2][0] \quad (4)$$

【0014】

以上の処理により、ソースの画素データ $C_{sp}[0][0]$ 、 $C_{sp}[1][0]$ 、 $C_{sp}[2][0]$ に、コンボリューションフィルタ係数 $C_v[0][0]$ 、 $C_v[1][0]$ 、 $C_v[2][0]$ をそれぞれ乗算し、その乗算値を積算した値が得られたことになる。

【0015】

次に、ステップS8に進み、変数mが2より小さいか否かが判定される。いまの場合、 $m=2$ であるから、NOの判定が行われ、ステップS10に進む。ステップS10においては、nが2より小さいか否かが判定される。いまの場合、 $n=0$ であるから、YESの判定が行われ、ステップS11に進む。ステップS11においては、変数nが1だけインクリメントされ、いまの場合、 $n=1$ とされる。

【0016】

その後、ステップS6に戻り、 $m=0$ と初期設定された後、ステップS7において、再び上記(1)式の演算が行われる。これにより、次の演算が行われる。

$$C_{dp}[1][1] = C_{dp}[1][1] + C_v[0][1] * C_{sp}[0][1] \quad (5)$$

【0017】

次に、ステップS8に進み、変数mが2より小さいか否かが判定され、いまの場合、 $m=0$ であるから、YESの判定が行われ、ステップS9で、 $m=1$ とされた後、ステップS7に戻り、再び(1)式の演算が行われ、次の(6)式が演算される。

$$C_{dp}[1][1] = C_{dp}[1][1] + C_v[1][1] * C_{sp}[1][1] \quad (6)$$

【0018】

以上の処理が繰り返される結果、以下に示す(7)式乃至(10)式の演算が行われる。

$$C_{dp}[1][1] = C_{dp}[1][1] + C_v[2][1] * C_{sp}[2][1] \quad (7)$$

$$C_{dp}[1][1] = C_{dp}[1][1] + C_v[0][2] * C_{sp}[0][2] \quad (8)$$

$$C_{dp}[1][1] = C_{dp}[1][1] + C_v[1][2] * C_{sp}[1][2] \quad (9)$$

$$C_{dp}[1][1] = C_{dp}[1][1] + C_v[2][2] * C_{sp}[2][2] \quad (10)$$

これにより、1個の画素データ $C_{sp}[1][1]$ を対象画素とするコンボリューションフィルタ処理が完了したことになる。

【0019】

このとき、ステップS10において、 $n=2$ となっているから、NOの判定が行われ、ステップS12に進む。ステップS12においては、変数 i が $HMAX-2$ （この例の場合、 $HMAX=6$ なので、 $HMAX-2=4$ より小さいか否かが判定される。いまの場合、 $i=1$ であり、 $HMAX-2$ より小さいからYESの判定が行われ、ステップS13に進み、変数 i が1だけインクリメントされて、 $i=2$ とされる。その後、ステップS5に戻り、それ以降の処理が実行される。すなわち、画素データ $C_{sp}[1][1]$ を対象画素とするコンボリューションフィルタ演算処理と同様に、画素データ $C_{sp}[2][1]$ を対象画素とするコンボリューションフィルタ演算が行われることになる。

【0020】

$j=1$ の行における画素データに対するコンボリューションフィルタ処理演算が順次行われ、画素データ $C_{sp}[1][HMAX-2]$ （図48の例の場合、 $C_{sp}[1][4]$ ）まで行われると、ステップS12において、NOの判定が行われ、ス

テップ S 1 4 に進む。ステップ S 1 4 においては、変数 j が $V_{MAX}-2$ (この例の場合、 $V_{MAX}=6$ であるので、 $V_{MAX}-2=4$ となる) より小さいか否かが判定される。いまの場合、 $j=1$ であるから、YES の判定が行われ、ステップ S 1 5 に進み、 j が 1 だけインクリメントされ、 $j=2$ とされる。その後、ステップ S 4 に戻り、それ以降の処理が実行される。すなわち、 $j=2$ の行における画素データに対するコンボリューションフィルタ処理演算が上述した場合と同様に実行される。

【0021】

以上のようにして、 j が $V_{MAX}-2$ の行の i が $H_{MAX}-2$ の列のソース画素データ $C_{SP}[H_{MAX}-2][V_{MAX}-2]$ に対するコンボリューションフィルタ係数演算処理が完了すると、ステップ S 1 4 において、NO の判定が行われ、コンボリューションフィルタ処理演算処理が完了することになる。

【0022】

CPU 5 は、以上のようにして得られた演算結果のデータをグラフィックス処理部 3 に供給する。グラフィックス処理部 3 は、CPU 5 より入力された 1 フレーム分の画像データをメモリ 2 1 に記憶する。表示データ生成部 2 2 は、この画像データを表示データに変換して、CRT 4 に出力して、表示させる。

【0023】

以上においては、コンボリューションフィルタ演算処理を CPU 5 により行わせるようにしたが、イメージ処理部 2 に実行させる場合もある。あるいはまた、このような処理を実行するための専用のハードウェアを別途用意し、それに実行させる場合もある。

【0024】

【発明が解決しようとする課題】

このように従来の画像処理装置においては、CPU 5、イメージ処理部 2、あるいは専用のハードウェアにより、所定の演算を行わせるようにしている。その結果、CPU 5 またはイメージ処理部 2 に対する負担が大きくなるか、または、専用のハードウェアが必要となり、構成が複雑になるばかりでなく、コスト高となる課題があった。

【0025】

本発明はこのような状況に鑑みてなされたものであり、簡単な構成かつ低コストの装置で、画像処理を実行することができるようにするものである。

【0026】

【課題を解決するための手段】

請求項1に記載の画像処理装置は、ソースの画像データを画素単位で記憶する第1の記憶手段と、デスティネーションの画像データを画素単位で記憶する第2の記憶手段と、第1の記憶手段に記憶されたソースの画像データに、画素単位で所定の演算を施して第2の記憶手段にデスティネーションの画像データとしてポリゴン単位で描画する動作を、所定の演算結果が得られるまで繰り返し行う描画手段とを備えることを特徴とする。

【0027】

請求項8に記載の画像処理方法は、ソースの画像の画像データを画素単位で記憶する第1の記憶手段と、デスティネーションの画像データを画素単位で記憶する第2の記憶手段とを備える画像処理装置の画像処理方法において、第1の記憶手段に記憶されたソースの画像データに、画素単位で所定の演算を施して第2の記憶手段にデスティネーションの画像データとしてポリゴン単位で描画する動作を、所定の演算結果が得られるまで繰り返し行う描画ステップを含むことを特徴とする。

【0028】

請求項9に記載の提供媒体は、ソースの画像の画像データを画素単位で記憶する第1の記憶手段と、デスティネーションの画像データを画素単位で記憶する第2の記憶手段とを備える画像処理装置に、第1の記憶手段に記憶されたソースの画像データに、画素単位で所定の演算を施して第2の記憶手段にデスティネーションの画像データとしてポリゴン単位で描画する動作を、所定の演算結果が得られるまで繰り返し行う描画ステップを含む処理を実行させるプログラムを提供することを特徴とする。

【0029】

請求項10に記載の画像処理装置は、ソースの画像データを画素単位で記憶す

る第1の記憶部と、デスティネーションの画像データを画素単位で記憶する第2の記憶部とを備える記憶手段と、第1の記憶部に記憶されたソースの画像データに、画素単位で所定の演算を施して第2の記憶部にデスティネーションの画像データとしてポリゴン単位で描画する動作を、所定の演算結果が得られるまで繰り返し行わせる描画コマンドを生成する生成手段と、生成手段により生成された描画コマンドを実行する実行手段とを備えることを特徴とする。

【0030】

請求項17に記載の画像処理方法は、ソースの画像データを画素単位で第1の記憶部に記憶させるとともに、デスティネーションの画像データを画素単位で第2の記憶部に記憶させる記憶ステップと、記憶ステップで、第1の記憶部に記憶されたソースの画像データに、画素単位で所定の演算を施して第2の記憶部にデスティネーションの画像データとしてポリゴン単位で描画する動作を、所定の演算結果が得られるまで繰り返し行わせる描画コマンドを生成する生成ステップとを含むことを特徴とする。

【0031】

請求項18に記載の提供媒体は、ソースの画像データを画素単位で第1の記憶部に記憶させるとともに、デスティネーションの画像データを画素単位で第2の記憶部に記憶させる記憶ステップと、記憶ステップで、第1の記憶部に記憶されたソースの画像データに、画素単位で所定の演算を施して第2の記憶部にデスティネーションの画像データとしてポリゴン単位で描画する動作を、所定の演算結果が得られるまで繰り返し行わせる描画コマンドを生成する生成ステップとを含む処理を実行させるプログラムを提供することを特徴とする。

【0032】

請求項19に記載の画像処理装置は、ソースの画像データを画素単位で記憶する第1の記憶手段と、デスティネーションの画像データを画素単位で記憶する第2の記憶手段と、第1の記憶手段に記憶されたソースの画像データに、画素単位で所定の演算のうちの一部の演算を施して第2の記憶手段にデスティネーションの画像データとしてポリゴン単位で描画する第1の描画手段と、第1の記憶手段に記憶されたソースの画像データに、画素単位で所定の演算のうちの一部の

演算を施して、第1の描画手段により既に描画されている画像データに加算または減算して、第2の記憶手段にデスティネーションの画像データとしてポリゴン単位で描画する第2の描画手段とを備えることを特徴とする。

【0033】

請求項26に記載の画像処理方法は、第1の記憶手段に記憶されたソースの画像データに、画素単位で所定の演算のうちの一部の演算を施して第2の記憶手段にデスティネーションの画像データとしてポリゴン単位で描画する第1の描画ステップと、第1の記憶手段に記憶されたソースの画像データに、画素単位で所定の演算のうちの一部の演算を施して、第1の描画ステップで既に描画されている画像データに加算または減算して、第2の記憶手段にデスティネーションの画像データとしてポリゴン単位で描画する第2の描画ステップとを含むことを特徴とする。

【0034】

請求項27に記載の提供媒体は、第1の記憶手段に記憶されたソースの画像データに、画素単位で所定の演算のうちの一部の演算を施して第1の記憶手段にデスティネーションの画像データとしてポリゴン単位で描画する第1の描画ステップと、第1の記憶手段に記憶されたソースの画像データに、画素単位で所定の演算のうちの一部の演算を施して、第1の描画ステップで既に描画されている画像データに加算または減算して、第2の記憶手段にデスティネーションの画像データとしてポリゴン単位で描画する第2の描画ステップとを含む処理を実行させるプログラムを提供することを特徴とする。

【0035】

請求項1に記載の画像処理装置、請求項8に記載の画像処理方法、および請求項9に記載の提供媒体においては、第1の記憶手段に記憶されたソースの画像データに、画素単位で所定の演算を施して第2の記憶手段にデスティネーションの画像データとしてポリゴン単位で描画する動作が、所定の演算結果が得られるまで繰り返し実行される。

【0036】

請求項10に記載の画像処理装置、請求項17に記載の画像処理方法、および

請求項 18 に記載の提供媒体においては、第 1 の記憶部に記憶されたソースの画像データに、画素単位で所定の演算を施して第 2 の記憶部にデスティネーションの画像データとしてポリゴン単位で描画する動作を、所定の演算結果が得られるまで繰り返し実行させる描画コマンドが生成される。

【0037】

請求項 19 に記載の画像処理装置、請求項 26 に記載の画像処理方法、および請求項 27 に記載の提供媒体においては、第 1 の記憶手段に記憶されたソースの画像データに、画素単位で所定の演算のうちの一部の演算を施して第 2 の記憶手段にデスティネーションの画像データとしてポリゴン単位で描画する動作が行われた後、さらに、第 1 の記憶手段に記憶された画像データに、画素単位で所定の演算のうち他の一部の演算を施して、既に描画されている画像データに加算または減算して、第 1 の記憶手段にデスティネーションの画像データとしてポリゴン単位で描画される。

【0038】

【発明の実施の形態】

以下に本発明の実施の形態を説明するが、特許請求の範囲に記載の発明の各手段と以下の実施の形態との対応関係を明らかにするために、各手段の後の括弧内に、対応する実施の形態（但し一例）を付加して本発明の特徴を記述すると、次のようになる。但し勿論この記載は、各手段を記載したものに限定することを意味するものではない。

【0039】

請求項 1 に記載の画像処理装置は、ソースの画像データを画素単位で記憶する第 1 の記憶手段（例えば、図 2 のテクスチャエリア 51）と、デスティネーションの画像データを画素単位で記憶する第 2 の記憶手段（例えば、図 2 の描画エリア 52）と、第 1 の記憶手段に記憶されたソースの画像データに、画素単位で所定の演算を施して第 2 の記憶手段にデスティネーションの画像データとしてポリゴン単位で描画する動作を、所定の演算結果が得られるまで繰り返し行う描画手段（例えば、図 1 のレンダリングエンジン 41）とを備えることを特徴とする。

【0040】

請求項3に記載の画像処理装置は、ソースの画像データと、デスティネーションの画像データとの間の演算モードを指定する指定手段（例えば、図1のメインCPU31）をさらに備えることを特徴とする。

【0041】

請求項10に記載の画像処理装置は、ソースの画像データを画素単位で記憶する第1の記憶部（例えば、図2のテクスチャエリア51）と、デスティネーションの画像データを画素単位で記憶する第2の記憶部（例えば、図2の描画エリア52）とを備える記憶手段（例えば、図1の画像メモリ43）と、第1の記憶部に記憶されたソースの画像データに、画素単位で所定の演算を施して第2の記憶部にデスティネーションの画像データとしてポリゴン単位で描画する動作を、所定の演算結果が得られるまで繰り返し行わせる描画コマンドを生成する生成手段（例えば、図1のメインCPU31）と、生成手段により生成された描画コマンドを実行する実行手段（例えば、図1のレンダリングエンジン41）とを備えることを特徴とする。

【0042】

請求項19に記載の画像処理装置は、ソースの画像データを画素単位で記憶する第1の記憶手段（例えば、図2のテクスチャエリア51）と、デスティネーションの画像データを画素単位で記憶する第2の記憶手段（例えば、図2の描画エリア52）と、第1の記憶手段に記憶されたソースの画像データに、画素単位で所定の演算のうちの一部の演算を施して第2の記憶手段にデスティネーションの画像データとしてポリゴン単位で描画する第1の描画手段（例えば、図7のステップS37、ステップS38、ステップS39）と、第1の記憶手段に記憶されたソースの画像データに、画素単位で所定の演算のうちの一部の演算を施して、第1の描画手段により既に描画されている画像データに加算または減算して、第2の記憶手段にデスティネーションの画像データとしてポリゴン単位で描画する第2の描画手段（例えば、図7のステップS37、ステップS40、ステップS41）とを備えることを特徴とする。

【0043】

請求項21に記載の画像処理装置は、ソースの画像データと、デスティネーションの画像データとの間の演算モードを指定する指定手段（例えば、図7のステップS34）をさらに備えることを特徴とする。

【0044】

図1は、本発明の画像処理装置を適用したコンピュータエンタテインメント装置の構成を示すブロック図である。メインCPU31には、バス34を介して、メインメモリ32と画像処理チップ33が接続されている。メインCPU31は、描画コマンドを生成し、画像処理チップ33の動作を制御する。メインメモリ32には、メインCPU31が各種の処理を実行する上において必要なプログラムやデータなどが適宜記憶される。

【0045】

画像処理チップ33のレンダリングエンジン41は、メインCPU31から供給される描画コマンドに対応して、メモリインターフェース42を介して、画像メモリ43に所定の画像データを描画する動作を実行する。メモリインターフェース42とレンダリングエンジン41の間には、バス45が接続されており、メモリインターフェース42と画像メモリ43の間には、バス46が接続されている。バス46は、例えば128ビットのビット幅を有し、レンダリングエンジン41は、画像メモリ43に対して、高速に描画処理を実行することができるようになされている。レンダリングエンジン41は、例えば、NTSC方式、あるいはPAL方式などの320×240画素の画像データ、あるいは、640×480画素の画像データを、リアルタイムに（1/30秒乃至1/60秒）の間に、十数回乃至数十回以上描画する能力を有している。

【0046】

画像処理チップ33はまた、プログラマブルCRTコントローラ（PCRTC）44を有しており、このPCRTC44は、ビデオカメラ35より入力された画像データの位置、大きさ、解像度などをリアルタイムに制御する機能を有している。PCRTC44は、ビデオカメラ35より入力された画像データを、メモリインターフェース42を介して、画像メモリ43のテクスチャエリア51（図2）に書き込む。

また、PCRTC 44 は、画像メモリ 43 の描画エリア 52（図 2）に描画された画像データをメモリインターフェース 46 を介して読み取り、これを CRT 36 に出力し、表示させるようになされている。

【0047】

画像メモリ 43 は、図 2 に示すように、テクスチャエリア 51 と描画エリア 52 を同一のエリアに指定できるユニファイドメモリ構造のものとされている。

【0048】

メインCPU 31 は、テクスチャエリア 51 の所定のテクスチャを描画エリア 52 に描画するとき、Flat_Texture_Polygon コマンドを生成し、これをレンダリングエンジン 41 に出力する。例えば、3 角形のポリゴンを描画するコマンドは次のようなコマンドとされる。

Flat_Texture_Triangle(Dx0,Dy0,Dx1,Dy1,Dx2,Dy2,Sx0,Sy0,Sx1,Sy1,Sx2,Sy2,L)

【0049】

ここで、Dx0,Dy0,Dx1,Dy1,Dx2,Dy2 は、デスティネーション（描画エリア 52）に描画する 3 角形の頂点の座標を表し、Sx0,Sy0,Sx1,Sy1,Sx2,Sy2 は、ソース（テクスチャエリア 51）の 3 角形の頂点の座標を表している。また、L は、点列（Sxn, Syn）で表現されるポリゴン（3 角形）内のテクスチャの画素値に乗算する輝度値を表している。

【0050】

同様に、4 角形描画コマンドは、次のように表される。

Flat_Texture_Rectangle(Dx0,Dy0,Dx1,Dy1,Dx2,Dy2,Dx3,Dy3,Sx0,Sy0,Sx1,Sy1,Sx2,Sy2,Sx3,Sy3,L)

【0051】

ここでも、Dx0,Dy0,Dx1,Dy1,Dx2,Dy2,Dx3,Dy3 は、デスティネーション（描画エリア 52）に描画する 4 角形の頂点の座標を表し、Sx0,Sy0,Sx1,Sy1,Sx2,Sy2,Sx3,Sy3 は、ソース（テクスチャエリア 51）の 4 角形の頂点の座標を表している。

【0052】

レンダリングエンジン 41 は、メインCPU 31 からバス 34 を介して、例えば

3 角形描画コマンドが入力されると、テクスチャエリア 51 の 3 つの頂点の座標 (Dx0, Dy0), (Dx1, Dy1), (Dx2, Dy2) で規定されるポリゴン内の点 (Sxn, Syn) の画素値に対して値 L を乗算し、その乗算した結果を座標 (Dx2, Dy2), (Sx0, Sy0), (Sx1, Sy1) で規定されるポリゴン内の対応する点 (Dxn, Dyn) に描画する。テクスチャエリア 51 の座標 (Sxn, Syn) に対応する描画エリア 52 上の座標 (Dxn, Dyn) は、次の (11) 式で表されるアフィン変換処理により求めることができる。

【0053】

【数 1】

$$\begin{pmatrix} D_{xn} \\ D_{yn} \end{pmatrix} = \begin{pmatrix} a, b \\ c, d \end{pmatrix} \begin{pmatrix} S_{xn} \\ S_{yn} \end{pmatrix} + \begin{pmatrix} OFX \\ OFY \end{pmatrix} \quad \cdots(11)$$

【0054】

なお、上記式における a 乃至 d は、回転のための係数であり、OFX, OFY は、平行移動のための係数である。

【0055】

実際には、描画エリア 52 においては、画素単位で描画が行われ、描画される画素値としては、その座標から、次の (12) 式で表される逆アフィン変換処理により求められるテクスチャエリア 51 上の座標の画素値が使用される。

【0056】

【数 2】

$$\begin{pmatrix} x+h \\ y+v \end{pmatrix} = \begin{pmatrix} a, b \\ c, d \end{pmatrix}^{-1} \begin{pmatrix} x-OFX \\ y-OFY \end{pmatrix} \quad \cdots(12)$$

【0057】

なお、上記式における h, v は、0 以上 1.0 未満の係数であり、x, y, X, Y は、整数である。

【0058】

また、通常、この座標値は、小数部を有し、図3に示すように、各画素の間に位置する座標となるため、次に示す式に従って、バイリニア補間した画素値 $SP(X, Y)$ が用いられる。

$$\begin{aligned} SP(X, Y) = & (1-h) * (1-v) * TP(x, y) \\ & + h * (1-v) * TP(x+1, y) \\ & + (1-h) * v * TP(x, y+1) \\ & + h * v * TP(x+1, y+1) \end{aligned} \quad (13)$$

【0059】

すなわち、テクスチャエリア51上の点 $TP(x, y)$ から x 軸方向に $+h$ 、 y 軸方向に $+v$ だけ離れた点 $(x+h, y+v)$ の画素値は、その周囲の点 $TP(x, y)$ 、 $TP(x+1, y)$ 、 $TP(x, y+1)$ 、 $TP(x+1, y+1)$ からの距離に対応して重み付けされた値が用いられる。

【0060】

さらに、レンダリングエンジン41は、メインCPU31からのブレンディングモード設定関数 $Set_Mode(MODE)$ により指定されたモードに対応して、描画エリア52上のデスティネーションの画素値 $DP(X, Y)$ とテクスチャエリア51上のソースの画素値 $SP(X, Y)$ との間で、ブレンディング処理を行わせる。

【0061】

レンダリングエンジン41が実行するブレンディングモードには、モード0乃至モード3が存在し、各モードにおいては、次のようなブレンディングが実行される。

MODE0 : $SP(X, Y)$

MODE1 : $DP(X, Y) + SP(X, Y)$

MODE2 : $DP(X, Y) - SP(X, Y)$

MODE3 : $(1 - \alpha_{SP}(X, Y)) * DP(X, Y)$

$+ \alpha_{SP}(X, Y) * SP(X, Y)$

【0062】

なお、 $\alpha_{SP}(X, Y)$ は、ソースの画素値の α 値を表している。

【0063】

すなわち、モード0においては、ソースの画素値が、そのままデスティネーションに描画され、モード1においては、ソースの画素値が、デスティネーションの画素値に加算して描画され、モード2においては、デスティネーションの画素値からソースの画素値を減算して描画が行われる。また、モード3においては、ソースの α 値に対応する重み付けを行って、ソースの画素値とデスティネーションの画素値が合成される。

【0064】

画像メモリ43の描画エリア52に描画された画像データは、メモリアンターフェース46を介して、PCRTC44に読み出され、そこからCRT36に出力され、表示される。

【0065】

次に、ビデオカメラ35より入力された画像データに対して、コンボリューションフィルタ処理を行った後、CRT36に出力する場合の処理について説明する。

【0066】

すなわち、この場合、図4に示すように、ビデオカメラ35より入力された処理対象の画像データが、PCRTC44からメモリアンターフェース42を介して、画像メモリ43のテクスチャエリア51に書き込まれる。これにより、例えば図5に示すように、テクスチャエリア51には、 $HMAX \times VMAX$ 個のソースの画素データ C_{sp} が書き込まれた状態となる。

【0067】

この画像データに対して、コンボリューションフィルタ演算を行うコンボリューションフィルタ係数は、例えば図6に示すように、 $m \times n$ 個（図6の例の場合、 3×3 個）用意される。このコンボリューションフィルタ係数を、ソースの対象画素を中心とする 3×3 個の画素データに乗算し、その乗算結果を加算した値を、対象画素に対応するデスティネーションの画素データとして描画エリア52上に描画する。

【0068】

但し、この実施の形態においては、 3×3 個のコンボリューションフィルタ係数

の中から、1つのコンボリューションフィルタ係数 C_{00} が最初に抽出され、このコンボリューションフィルタ係数 C_{00} が、最初にテクスチャエリア51上のソースの全ての画素データに対して乗算され、描画される。そして、次に、1画素分だけ移動した位置のコンボリューションフィルタ係数、例えば C_{01} が読み出され、このコンボリューションフィルタ係数がソースの画素データ全てに対して乗算され、これが既に演算値が描画されている描画エリア52上のデスティネーションとしての画素値に加算または減算して描画される。

【0069】

以上のような処理が繰り返し実行されて、コンボリューションフィルタ演算処理が実行されることになる。

【0070】

すなわち、コンボリューションフィルタ係数が正の場合、メインCPU31は、レンダリングエンジン41に対して、次に示すようなコマンドを出力して、コンボリューションフィルタ処理を実行させる。

【0071】

```
Set_Mode(0);
Flat_Rectangle(1,1,HMAX-1,1,HMAX-1,VMAX-1,1,VMAX-1,    0);
Set_Mode(1);
for(n=0; n< 3; n++) {
    for(m=0; m< 3; m++) {
        Flat_Texture_Rectangle(1,1,HMAX-1, 1,HMAX-1, VMAX-1, 1,VMAX-1,
                                m,n,HMAX-2+m,n,HMAX-2+m,VMAX-2+n,m,VMAX-2+n,
                                Cv[m] [n] );
    }
}
```

【0072】

以上の処理をフローチャートに示すと、図7に示すようになる。

【0073】

すなわち、最初にステップS31において、メインCPU31は、レンダリングエンジン41に、デスティネーションとしての描画エリア52（出力領域）をクリアさせた後、コンボルーションフィルタ係数 $C_v[m][n]$ を所定の値に設定させる。いまの場合、図6の示すように、 3×3 個のコンボルーションフィルタ係数が用意されることになる。

【0074】

次に、ステップS32において、メインCPU31は、レンダリングエンジン41に対して、ブレンディングモードとしてモード0を設定させるコマンドを出力する。レンダリングエンジン41は、このコマンドに対応して、ブレンディングモードをモード0に設定する。

【0075】

次に、メインCPU31は、ステップS33において、 $HMAX \times VMAX$ 個の画素で構成されるデスティネーションのうち、4角形（1, 1, $HMAX-1$, 1, $HMAX-1$, $VMAX-1$, 1, $VMAX-1$ ）に、画素値0を描画させるコマンドを生成し、レンダリングエンジン41に出力する。レンダリングエンジン41は、このコマンドに対応して、図8に示すように、 $HMAX \times VMAX$ のデスティネーションのうち、 $j=0$ の行、並びに $i=0$ の列を除く各画素に、値0が描画される。

【0076】

次に、ステップS34において、レンダリングエンジン41は、メインCPU31からの指令に対応して、ブレンディングモードとしてモード1を設定する。すなわち、ソースの画素値を、デスティネーションの画素値に加算して、描画するモードが設定される。

【0077】

次に、ステップS35とS36において、図6に示すように、コンボルーションフィルタ係数の y 軸方向の座標を表す変数 n と、 x 軸方向の座標を表す変数 m が、それぞれ0に初期設定される。

【0078】

次に、ステップS37において、レンダリングエンジン41は、図5に示すよ

うな、 $HMAX \times VMAX$ 個の画素データ C_{SP} で構成されるソースの画素値のうち、4角形のポリゴン ($m, n, HMAX-2+m, n, HMAX-2+m, VMAX-2+n, m, VMAX-2+n$) の範囲内の画素値に、図6に示すようなコンボルーションフィルタ係数 $C_v[m][n]$ の値を乗算し、デスティネーションの4角形 ($1, 1, HMAX-1, 1, HMAX-1, VMAX-1, 1, VMAX-1$) の範囲内に描画する処理が実行される。これにより、いまの場合、画像メモリ43の描画エリア52の、図9に示すような、 $HMAX \times VMAX$ のデスティネーションエリアのうち、 $i = 1$ 乃至 $HMAX-1$ (図9の例の場合、 $HMAX-1 = 5$)、および $j = 1$ 乃至 $VMAX-1$ (図9の例の場合、 $VMAX-1 = 5$) の範囲に、図5に示すソースの $i = 0$ 乃至 4 と、 $j = 0$ 乃至 4 の範囲の画素データ C_{SP} に、係数 $C_{00} (= C_v[0][0])$ が乗算された値が描画される。例えば、

$$C_{dp11} = C_{SP00} \times C_{00}$$

となり、

$$C_{dp21} = C_{SP10} \times C_{00}$$

となる。また、

$$C_{dp12} = C_{SP01} \times C_{00}$$

となり、

$$C_{dp22} = C_{SP11} \times C_{00}$$

となる。

【0079】

次に、ステップS38に進み、変数 m が2より小さいか否かが判定される。いまの場合、 $m = 0$ であるから、YESの判定が行われ、ステップS39に進み、変数 m が1だけインクリメントされて、 $m = 1$ とされる。

【0080】

次に、ステップS37に戻り、図5の $i = 1$ 乃至 5 と、 $j = 0$ 乃至 4 で規定される範囲の画素値 C_{SP} に、コンボルーションフィルタ係数 $C_v[1][0] (= C_{10})$ を乗算した値が、 $i = 1$ 乃至 5 と $j = 1$ 乃至 5 で規定されるデスティネーションの4角形に、既に描画された値に加算して描画される。すなわち、図10に示すように、例えば、

$$C_{dp11} = C_{SP00} \times C_{00} + C_{SP10} \times C_{10}$$

となり、

$$C_{dp21} = C_{SP10} \times C_{00} + C_{SP20} \times C_{10}$$

となる。また、

$$C_{dp12} = C_{SP01} \times C_{00} + C_{SP11} \times C_{10}$$

となり、

$$C_{dp22} = C_{SP11} \times C_{00} + C_{SP21} \times C_{10}$$

となる。

【0081】

次に、ステップS38に進み、変数mが2より小さいか否かが判定され、いまの場合、m=1であるから、YESの判定が行われ、ステップS39に進み、m=2とされる。その後、ステップS37に進み、ソースの4角形のi=2乃至5と、j=1乃至5で規定される範囲の画素値に、コンボリューションフィルタ係数 $C_v[2][0]$ (= C_{20})を乗算した値が、既に描画されているデスティネーションの4角形の画素値に、さらに加算して描画される。

【0082】

これにより、例えば、図11に示すように、 C_{dp11} 、 C_{dp} 、 C_{21} 、 C_{dp12} 、 C_{dp22} は、次のような値となる。

$$C_{dp11} = C_{SP00} \times C_{00} + C_{SP10} \times C_{10} + C_{SP20} \times C_{20}$$

$$C_{dp21} = C_{SP10} \times C_{00} + C_{SP20} \times C_{10} + C_{SP30} \times C_{20}$$

$$C_{dp12} = C_{SP01} \times C_{00} + C_{SP11} \times C_{10} + C_{SP21} \times C_{20}$$

$$C_{dp22} = C_{SP11} \times C_{00} + C_{SP21} \times C_{10} + C_{SP31} \times C_{20}$$

【0083】

なお、この場合、ソースの画素データとして、 C_{SP6j} の画素値が必要となるが、このような画像データは実際に存在しないので、図11に示すように、i=5(=HMAX-1)の画素データ C_{dp} は、無効の画素データとされる。

【0084】

次に、ステップS38に進み、mが2より小さいか否かが判定される。いまの場合、m=2であるから、NOの判定が行われ、ステップS40に進む。ステッ

ブ S 4 0 においては、 n が 2 より小さいか否かが判定される。いまの場合、 $n = 0$ であるから、YES の判定が行われ、ステップ S 4 1 に進み、変数 n が 1 だけインクリメントされ、 $n = 1$ とされた後、ステップ S 3 6 に戻る。

【0085】

ステップ S 3 6 において、変数 m が再び 0 に初期設定され、ステップ S 3 7 において、 $i = 0$ 乃至 4 と、 $j = 1$ 乃至 4 で規定されるの範囲のソースの画素データ C_{dp} にコンボリューションフィルタ係数 $C_v [0] [1] (= C_{01})$ の値を乗算し、既に描画されているデスティネーションの値にさらに加算して描画する処理が実行される。

【0086】

これにより、例えば、図 1 2 に示すように、 C_{dp11} , C_{dp} , C_{21} , C_{dp12} , C_{dp22} は、次のような値となる。

$$C_{dp11} = C_{SP00} \times C_{00} + C_{SP10} \times C_{10} + C_{SP20} \times C_{20} + C_{SP01} \times C_{01}$$

$$C_{dp21} = C_{SP10} \times C_{00} + C_{SP20} \times C_{10} + C_{SP30} \times C_{20} + C_{SP11} \times C_{01}$$

$$C_{dp12} = C_{SP01} \times C_{00} + C_{SP11} \times C_{10} + C_{SP21} \times C_{20} + C_{SP02} \times C_{01}$$

$$C_{dp22} = C_{SP11} \times C_{00} + C_{SP21} \times C_{10} + C_{SP31} \times C_{20} + C_{SP12} \times C_{01}$$

【0087】

以下、同様の処理が、ステップ S 4 0 において、変数 n が 2 より小さくないと判定されるまで ($n = 2$ となるまで)、繰り返し実行される。その結果、デスティネーションエリアには、図 1 3 に示すような描画が行われる。同図に示すように、例えば、 C_{dp11} , C_{dp21} , C_{dp12} , C_{dp22} の値は、それぞれ次のようになる。

【0088】

$$C_{dp11} = C_{SP00} \times C_{00} + C_{SP10} \times C_{10} + C_{SP20} \times C_{20} + C_{SP01} \times C_{01} \\ + C_{SP11} \times C_{11} + C_{SP21} \times C_{21} + C_{SP02} \times C_{02} + C_{SP12} \times C_{12} + C_{SP22} \times C_{22}$$

$$C_{dp21} = C_{SP10} \times C_{00} + C_{SP20} \times C_{10} + C_{SP30} \times C_{20} + C_{SP11} \times C_{01} \\ + C_{SP21} \times C_{11} + C_{SP31} \times C_{21} + C_{SP12} \times C_{02} + C_{SP22} \times C_{12} + C_{SP32} \times C_{22}$$

$$C_{dp12} = C_{SP01} \times C_{00} + C_{SP11} \times C_{10} + C_{SP21} \times C_{20} + C_{SP02} \times C_{01} \\ + C_{SP12} \times C_{11} + C_{SP22} \times C_{21} + C_{SP03} \times C_{02} + C_{SP13} \times C_{12} + C_{SP23} \times C_{22}$$

$$C_{dp22} = C_{SP11} \times C_{00} + C_{SP21} \times C_{10} + C_{SP31} \times C_{20} + C_{SP12} \times C_{01} \\ + C_{SP22} \times C_{11} + C_{SP32} \times C_{21} + C_{SP13} \times C_{02} + C_{SP23} \times C_{12} + C_{SP33} \times C_{22}$$

【0089】

なお、この場合、ソースの画素データとして、 C_{SPi6} の画素値が必要となるが、このような画像データは実際に存在しないので、図11に示すように、 $j = 5$ ($=VMAX-1$)の画素データ C_{dp} は、無効の画素データとされる。

【0090】

以上のようにして、 $i = 1$ 乃至 $HMAX-2$ と、 $j = 1$ 乃至 $VMAX-2$ で規定される範囲に、コンボリューションフィルタ処理を実行した画素データが得られる。

【0091】

以上の処理は、コンボリューションフィルタ係数が正の場合の処理であるが、コンボリューションフィルタ係数に負数が存在する場合には、メインCPU31からレンダリングエンジン41に出力する描画コマンドは、例えば、次に示すようになる。

【0092】

```
Set_Mode(0);
Flat_Rectangle(1,1,HMAX-1,1,HMAX-1,VMAX-1,1,VMAX-1, CMAX/2);
for(n=0; n< 3; n++) {
    for(m=0; m< 3; m++) {
        if(Cv[m][n]>0) {
            Set_Mode(1);
            Flat_Texture_Rectangle(1,1,HMAX-1, 1,HMAX-1, VMAX-1, 1,VMAX-1,m
,
            n,HMAX-2+m,n,HMAX-2+m,VMAX-2+n,m,VMAX-2+n, Cv[m][n]);
        }
        if(Cv[m][n]<0) {
            Set_Mode(2);
            Flat_Texture_Rectangle(1,1,HMAX-1, 1,HMAX-1, VMAX-1, 1,VMAX-1,m
,
            n,HMAX-2+m,n,HMAX-2+m,VMAX-2+n,m,VMAX-2+n, -Cv[m][n]);
        }
    }
}
```

```

    }
}

Set_Mode(2);

Flat_Rectangle(1,1,HMAX-1,1,HMAX-1,VMAX-1,1,VMAX-1, CMAX/2);

```

【0093】

また、この処理をフローチャートに表すと、図14と図15に示すようになる。

【0094】

すなわち、最初にステップS51において、コンボリューションフィルタ係数 $C_v[m][n]$ が所定の値に設定される。次に、ステップS52において、レンダリングエンジン41は、メインCPU31からの指令に対応してブレンディングモードとしてモード0を設定する。そして、ステップS53において、レンダリングエンジン41は、画像メモリ43の描画エリア52のデスティネーションエリアの4角形(1, 1, HMAX-1, 1, HMAX-1, VMAX-1, 1, VMAX-1)に、 $CMAX/2$ を描画する。 $CMAX$ は、画素値の最大値を意味する。すなわち、この場合、図16に示すように、デスティネーションエリアの $i = 1$ 乃至 $HMAX-1$ と、 $j = 1$ 乃至 $VMAX-1$ で規定される範囲に、画素値の最大値の $1/2$ の値が描画される。

【0095】

次に、ステップS54とステップS55において、変数 n と変数 m に、それぞれ0が初期設定され、ステップS56において、コンボリューションフィルタ係数 $C_v[m][n]$ が正であるか否かが判定される。このフィルタ係数が正である場合には、ステップS57に進み、ブレンディングモードとしてモード1が設定される。次に、ステップS58に進み、描画処理が行われる。この描画処理は、上述した図7のステップS37における場合と同様の処理であるので、その説明は省略する。

【0096】

ステップS58の処理の次に、ステップS59に進み、変数 m が2より小さいか否かが判定され、 m が2より小さいと判定された場合、ステップS60に進み

、変数 m が 1 だけインクリメントされ、ステップ S 56 に戻る。そして、それ以降の処理が繰り返し実行される。

【0097】

ステップ S 59 において、変数 m が 2 より小さくないと判定された場合、すなわち、 m が 2 と等しい場合、ステップ S 61 に進み、変数 n が 2 より小さいか否かが判定される。変数 n が 2 より小さい場合、ステップ S 62 に進み、変数 n が 1 だけインクリメントされた後、ステップ S 55 に戻り、それ以降の処理が繰り返し実行される。

【0098】

ステップ S 61 において、変数 n が 2 より小さくないと判定された場合（変数 n が 2 と等しい場合）、ステップ S 63 に進み、ブレンディングモードとしてモード 2 が設定される。そして、ステップ S 64 において、レンダリングエンジン 41 は、デスティネーションエリアの 4 角形（1, 1, HMAX-1, 1, HMAX-1, VMAX-1, 1, VMAX-1）の画素値から、ステップ S 53 で描画した CMAX/2 の値を減算して、描画する処理が実行される。

【0099】

一方、ステップ S 56 において、コンボルーションフィルタ係数 $C_v[m][n]$ が正ではないと判定された場合、ステップ S 65 に進み、この係数が負であるか否かが判定される。この係数が負であると判定された場合、ステップ S 66 に進み、ブレンディングモードとしてモード 2 が設定される。そして、ステップ S 67 において、ステップ S 58（すなわち、図 7 のステップ S 37）における場合と同様の描画処理が実行される。

【0100】

ステップ S 65 において、係数 $C_v[m][n]$ が負ではないと判定された場合、すなわち、この係数が 0 であると判定された場合、ステップ S 66 とステップ S 67 の処理はスキップされる。

【0101】

そして、ステップ S 67 の処理の次に、ステップ S 59 に進み、上述した場合と同様の処理が実行される。

【0102】

以上においては、いわば、画素データをポリゴン単位で描画することにより、コンボルーションフィルタ演算処理を実行するようにしたのであるが、画素単位で描画することでも、同様の処理を実行することが可能である。図17は、この場合の処理例を表している。

【0103】

すなわち、最初にステップS71において、コンボルーションフィルタ係数 $C_v[m][n]$ が設定される。ステップS72においては、ブレンディングモードとしてモード1が設定される。そして、ステップS73とステップS74において、変数 j と変数 i が、それぞれ1に初期設定され、ステップS75において、デスティネーションの画素データ $C_{dp}[i][j]$ の値として、0が初期設定される。

【0104】

次に、ステップS76とステップS77において、変数 n と変数 m が0に初期設定される。ステップS78においては、ソースの画素データ $C_{sp}[i+m-1][j+n-1]$ にコンボルーションフィルタ係数 $C_v[m][n]$ を乗算し、デスティネーションの画素データ $C_{dp}[i][j]$ に加算して描画する処理が実行される。

【0105】

次に、ステップS79において、変数 m が2より小さいか否かが判定される。いまの場合、変数 $m=0$ であるのでYESの判定が行われ、ステップS80において、変数 m が1だけインクリメントされ、 $m=1$ とされた後、ステップS78に戻る。

【0106】

ステップS78において、 x 軸方向に1画素分だけ右隣のソースの画素データ $C_{sp}[i+m-1][j+n-1]$ に、 x 軸方向の右隣のコンボルーションフィルタ係数 $C_v[m][n]$ を乗算し、同一のデスティネーションの画素データ $C_{dp}[i][j]$ に加算して描画する処理が実行される。

【0107】

同様の処理が、ステップS79において、 m が2より小さくない（2と等しい）と判定されるまで、繰り返し実行される。そして、ステップS81に進み、変数 n が2より小さいか否かが判定され、2より小さい場合には、ステップS82に進み、変数 n が1だけインクリメントされた後、ステップS77に戻り、それ以降の処理が繰り返し実行される。

【0108】

以上の処理が、ステップS81において、変数 n が2より小さくない（2と等しい）と判定されるまで繰り返し実行される。これにより、同一のデスティネーションの画素データ $C_{dp}[i][j]$ に、 3×3 のコンボリューションフィルタ係数が乗算された結果が、繰り返し描画され、加算される。すなわち、これにより、1つの対象画素のコンボリューションフィルタ演算処理が完了したことになる。

【0109】

次に、ステップS83に進み、 i が $HMAX-2$ より小さいか否かが判定され、小さいと判定され場合、ステップS84に進み、変数 i が1だけインクリメントされた後、ステップS75に戻り、それ以降の処理が繰り返し実行される。すなわち、 x 軸方向にデスティネーションの画素データが1ずつ移動され、同様の処理が実行される。

【0110】

ステップS83において、変数 i が $HMAX-2$ より小さくないと判定された場合（等しいと判定された場合）、ステップS85に進み、変数 j が $VMAX-2$ より小さいか否かが判定される。変数 j が $VMAX-2$ より小さいと判定された場合、ステップS86に進み、変数 j が1だけインクリメントされた後、ステップS74に戻り、それ以降の処理が繰り返し実行される。すなわち、同様の処理が、デスティネーションの画素データを y 方向に移動しながら、順次、繰り返し実行される。そして、ステップS85において、 $j = VMAX-2$ と判定されたとき、コンボリューションフィルタ処理が終了される。

【0111】

このようにしても、図7または図14と図15に示した場合と同様の結果を得

ることが可能である。しかしながら、このように、ポリゴン単位ではなく、画素単位で描画処理を行うと、アドレスの発生処理に時間がかかるため、結局、描画処理に時間がかかることになる。従って、図7または図14と図15に示すように、ポリゴン単位で描画処理を実行するのが好ましい。

【0112】

次に、ピラミッドフィルタ処理について説明する。このピラミッドフィルタ処理においては、図18に示すように、処理画像の相互に隣接する4個の画素値の平均値を求め、その画素を4個の画素の中心に配置する処理が繰り返される。すなわち、バイリニア補間により、近傍4点の平均画素値を演算する処理を実行すると、 $n \times n$ 個（ n は2のべき乗の値）の処理画像から、 $(n/2) \times (n/2)$ の画像データが得られる。この処理を繰り返し実行すると、最終的には、ピラミッドの頂点の1画素のデータは、ピラミッドの底面の全画素の平均値を表す画素データとなる。

【0113】

このようなピラミッドフィルタ処理を実行する場合、メインCPU31は、レンダリングエンジン41に対して、次のような描画コマンドを出力する。

【0114】

```
int L; /*ソースエリアの一辺の長さ*/
int offset;
L=2N^ ; /*初期画像の一辺の長さ*/
offset=0;
while(L>1){
    Set_Texture_Base(0,offset); /*テクスチャエリアのベースポイント設定
*/
    offset += L;
    Set_Drawing_Base(0,offset); /*描画エリアのベースポイント設定*/
    Flat_Texture_Rectangle(0,0, L/2,0, L/2,L/2, 0,L/2, 0.5,0.5, L+0.5,
0.5, L+0.5,L+0.5, 0.5,L+0.5, 1.0);
    L=L/2;
```

}

【0115】

この描画コマンドをフローチャートに表すと、図19に示すようになる。最初に、ステップS91において、変数offsetに0が初期設定される。次に、ステップS92において、テクチャエリア51のベースポイントを(0, offset)に設定する処理が実行される。すなわち、図20に示すように、ベースポイントT(0, 0)が設定される。次に、ステップS93に進み、変数offsetがLだけインクリメントされる。そして、ステップS94において、描画エリア52のベースポイントに(0, offset)が設定される。いまの場合、図20に示すように、ベースポイントD(0, L)が設定される。

【0116】

次に、ステップS95において、ソース(テクチャエリア)の4角形(0.5, 0.5, L+0.5, 0.5, L+0.5, L+0.5, 0.5, L+0.5)の画素値に1を乗算して、デスティネーションの4角形(0, 0, L/2, 0, L/2, L/2, 0, L/2)に加算して描画する処理が実行される。すなわち、これにより、図18に示す最も下の(ピラミッドの底面の)処理画像から、1つ上の階層の処理画像が得られたことになる。

【0117】

次に、ステップS96に進み、変数Lが現在の値の1/2とされる。ステップS97において、変数Lが1より大きいかが判定され、変数Lが1より大きい場合には、ステップS92に戻り、それ以降の処理が繰り返し実行される。すなわち、これにより、第2番目の階層から、さらに第3番目の階層の画像データが得られることになる。

【0118】

以下、同様の処理が繰り返し実行され、ステップS97において、変数Lが1より大きくないと判定された場合(変数Lが1に等しいと判定された場合)、ピラミッドフィルタ処理が終了される。

【0119】

次に、フレーム間差分処理について説明する。このフレーム間差分処理におい

ては、図21に示すように、時刻 t におけるフレームの画像と、時刻 $t+1$ におけるフレームの画像の差が演算される。これにより、動きのある画像のエリアを抽出することができる。

【0120】

すなわち、この場合、メインCPU31は、レンダリングエンジン41に対して、図22のフローチャートに示すような処理を実行させる。最初にステップS101において、レンダリングエンジン41は、メインCPU31からの指令に対応して、ブレンディングモードとしてモード2を設定する。次に、ステップS102において、レンダリングエンジン41は、ビデオカメラ35より入力された画像データのうち、時間的に後のフレームの画像データをデスティネーションの画像とし、時間的に前のフレームの画像データをソース画像データとする。そして、ステップS103において、レンダリングエンジン41は、デスティネーションの4角形の画素値から、ソースの4角形の画素値を減算して描画する処理を実行する。デスティネーションエリアのフレームの画素データとソースエリアのフレームの画素データは、静止画領域においては、その値は実質的に等しい値となる。その結果、ステップS103における処理を実行すると、その画素データの値は、ほぼ0となる。

【0121】

これに対して、動きのある領域の画素データの値は、デスティネーションにおける場合とソースにおける場合とで異なる値となる。従って、ステップS103における処理の結果得られる画像データの値は、0以外の所定の大きさを有する値となる。そこで、フレーム間差分の画像データの各画素データの値の大きさから、動き領域と静止画領域とを区分することができる。

【0122】

次に、画像間距離について説明する。画像間距離は、図23に示すように、画像Aと画像Bの2つのフレームの画像の異なる程度を表すものである。この画像間距離を求める場合、画像Aと画像Bの差分画像を求める処理が、クランプ付処理で実行される。クランプ付処理とは、0より小さい値は0に、また、最大値より大きい値は最大値に、飽和する処理を意味する。画像Aと画像Bの画像間距離

を求める場合、画像Aから画像Bを減算したクランプ付差分画像データと、画像Bから画像Aを減算したクランプ付差分画像データが求められ、それらを加算することで、絶対値化差分画像データが得られる。

【0123】

例えば、画像Aの所定の画素データの値が13であり、画像Bの対応する画素データの値が20であったとすると、 $A-B$ の値は-7となるが、クランプされる結果、その値は0となる。また、 $B-A$ の値は7となる。その結果、両者を加算した値は7となる。

【0124】

このようにして、絶対値化差分画像データが得られたとき、次に、この絶対値化差分画像データのピラミッドフィルタ処理が実行される。上述したように、ピラミッドフィルタ処理を実行すると、その頂点の1つの画素の値は、処理画像（絶対値化差分画像）の全画素の平均値となる。この平均値は、画像Aと画像Bの差分の絶対値の平均値であるから、画像Aと画像Bの画像間距離と考えることができる。

【0125】

メインCPU31は、画像間距離を求める場合、レンダリングエンジン41に、図24のフローチャートに示すような処理を実行させる。最初に、ステップS111において、レンダリングエンジン41は、第1の画像（図23における画像A）をデスティネーションの画像とし、第2の画像（画像B）をソースの画像とする。ステップS112において、レンダリングエンジン41は、ブレンディングモードをモード2に設定する。次に、ステップS113において、レンダリングエンジン41は、デスティネーションの画像（画像A）の画素値から、ソースの画像（画像B）の画素値を減算し、クランプして描画する。これにより、第1の距離画像領域に、 $A-B$ のクランプ付き差分画像データが得られたことになる。

【0126】

次に、ステップS114において、第2の画像（画像B）をデスティネーション画像とし、第1の画像（画像A）をソース画像とする処理が実行される。次に

、ステップS115において、ブレンディングモード2が設定される。ステップS116においては、デスティネーションの画像（画像B）の画素値から、ソースの画像（画像A）の画素値を減算し、クランプして描画する処理が実行される。これにより、第2の距離画像領域に、 $B - A$ のクランプ付き差分画像データが得られたことになる。

【0127】

次に、ステップS117において、レンダリングエンジン41は、第1の距離画像領域の画像（ $A - B$ のクランプ付き差分画像）をデスティネーションとし、第2の距離画像領域の画像（ $B - A$ のクランプ付き差分画像）をソース画像とする。そして、ステップS118において、ブレンディングモードとしてモード1が設定される。ステップS119においては、デスティネーションの画像（ $A - B$ のクランプ付き差分画像）の画素値に、ソースの画像（ $B - A$ のクランプ付き差分画像）の画素値を加算して、描画する処理が実行される。これにより、絶対値化差分画像データが得られたことになる。

【0128】

次に、ステップS120において、図19に示したピラミッドフィルタ処理を絶対値化差分画像に施すことにより、画像間距離を生成することができる。

【0129】

次に、パターンマッチング処理について説明する。対象画像とテンプレート画像とのパターンマッチング処理を実行する場合、テンプレートの画像の大きさが大きい場合（例えば、対象画像の大きさと等しい場合）、メインCPU31は、レンダリングエンジン41に対して、例えば図25のフローチャートに示すような処理を実行させる。

【0130】

すなわち、最初にステップS131において、レンダリングエンジン41は、 $P \times Q$ 画素の対象画像を、 $p \times q$ 画素までピラミッドフィルタ処理演算により縮小させる。同様に、ステップS132において、レンダリングエンジン41は、 $P \times Q$ 画素のテンプレート画像を、 $p \times q$ 画素までピラミッドフィルタ処理により縮小させる。

【0131】

次に、ステップS133において、レンダリングエンジン41は、 $p \times q$ 画素の対象画像と、 $p \times q$ 画素のテンプレートの画像の $s \times t$ 画素のブロックの画像間距離を求める。この画像間距離の演算は、上述した図24のフローチャートに示すように行われる。

【0132】

次に、ステップS134において、すべての範囲で画像間距離を求めたか否かが判定され、まだ画像間距離を求めている範囲が存在する場合には、ステップS135に進み、ブロックの位置を1画素分ずらす処理が実行される。そして、ステップS133に戻り、同様の処理が繰り返し実行される。

【0133】

ステップS134において、すべての範囲で画像間距離を求めたと判定された場合、ステップS136に進み、メインCPU31は、レンダリングエンジン41を制御し、ステップS133の処理を繰り返し実行することにより求めた画像間距離のうち、最も短い距離の領域の近傍の、より詳細な画像を抽出させる。

【0134】

次に、ステップS137に進み、ステップS136で抽出した領域の $s \times t$ 画素のブロックの画像間距離を求める処理が実行される。ステップS138においては、すべての範囲で画像間距離を求めたか否かが判定され、まだ画像間距離を求めている範囲が存在する場合には、ステップS139に進み、ブロックの位置を1画素分ずらす処理が実行される。その後、ステップS137に戻り、同様の処理が繰り返し実行される。

【0135】

ステップS138において、すべての範囲で画像間距離を求めたと判定された場合、ステップS140に進み、ステップS137の処理を繰り返し実行することにより得られた複数の画像間距離のうち、最も短い距離のものに対応するブロックが抽出される。

【0136】

以上においては、対象画像に対するテンプレートの画像の大きさが比較的大き

い場合の処理であるが、テンプレートが対象画像に較べて、充分小さい場合には、図 26 のフローチャートに示すように、パターンマッチング処理を行うことができる。

【0137】

すなわち、この場合においては、ステップ S 151 において、第 1 の画像として対象画像が選択される。そして、ステップ S 152 において、レンダリングエンジン 41 は、図 27 に示すように、比較的小さいテンプレートを複数個配列することにより、第 2 の画像を生成する。ステップ S 153 においては、ステップ S 151 において選択された対象画像と、ステップ S 152 において生成されたテンプレートの画像の画像間距離を演算する処理が実行される。この画像間距離の演算処理は、図 24 のフローチャートに示した場合と同様に実行されるが、この実施の形態の場合、図 27 に示すように、第 2 の画像が複数のテンプレートを配列することで生成されているので、このテンプレート単位で画像間距離が演算される。

【0138】

ステップ S 154 においては、すべての探索範囲で画像間距離を求めたか否かが判定され、まだ画像間距離を求めている範囲が存在する場合には、ステップ S 155 に進み、第 2 の画像の位置が 1 画素分だけずらされる。そして、ステップ S 153 に戻り、それ以降の処理が繰り返し実行される。

【0139】

ステップ S 154 において、すべての探索範囲で画像間距離を求めたと判定された場合、ステップ S 156 に進み、ステップ S 153 の処理を繰り返し実行することにより得られた複数の画像間距離の中から、最も短い画像間距離のブロックが抽出される。

【0140】

図 28 は、動きベクトルを検出する場合の処理例を表している。最初に、ステップ S 161 において、PCRTC 44 は、ビデオカメラ 35 より出力された前フレームの画像と現フレームの画像を、図 29 に示すように、画像メモリ 43 に記憶させる。次に、ステップ S 162 に進み、メイン CPU 31 は、前フレームの画像

の1つのブロックをテンプレートとして抽出する。そして、ステップS163において、メインCPU31は、ステップS162で抽出したテンプレートと、現フレームの対応する範囲の画像（ブロック）との画像間距離を求める処理を、図24のフローチャートを参照して説明したように実行する。

【0141】

次に、ステップS164において、メインCPU31は、図30に示すように、現フレームの探索範囲内のすべてについて探索したか否かを判定し、まだ探索していない範囲が存在する場合には、ステップS165に進み、現フレームの対応する範囲の画像（ブロック）の位置を1画素分だけずらす処理を実行する。そして、ステップS163に戻り、再びテンプレートの画像と現フレームのブロックの画像の画像間距離を求める処理が実行される。

【0142】

以上のような処理が、ステップS164において、すべての探索範囲内において探索が行われたと判定されるまで繰り返し実行される。ステップS164において、すべての探索範囲において探索が行われたと判定された場合、ステップS166に進み、メインCPU31は、ステップS163の処理を繰り返し実行することで求めた複数の画像間距離の中から、最短の画像間距離を求め、それに対応する現フレーム上のブロックを最短ブロックとして選択する。そして、ステップS167において、メインCPU31は、テンプレートと、ステップS166で選択した最短ブロックとの間の動きベクトルを求める。

【0143】

次に、ステップS168に進み、前フレームのすべてのブロックについて動きベクトルを求めたか否かをメインCPU31は判定する。まだ動きベクトルを求めているブロックが残っている場合には、ステップS162に戻り、前フレームから新たな1つのブロックをテンプレートとして抽出し、同様の処理が繰り返し実行される。

【0144】

ステップS168において、前フレームのすべてのブロックについて動きベクトルを求めたと判定された場合、処理は終了される。

【0145】

以上のようにして、動きベクトルを求めることが可能であるが、例えば図31に示すように、動きベクトルを求めることもできる。この例では、ステップS171において、ビデオカメラ35が出力した前フレームの画像と現フレームの画像が画像メモリ43に記憶される。次に、ステップS172において、図32に示すように、前フレームと現フレームの各ブロック毎の画像間距離が同時に求められる。

【0146】

さらに、ステップS173において、メインCPU31は、現フレームの前フレームに対する相対的位置を、全探索範囲内においてずらしたか否かを判定する。まだ全探索範囲内に渡って現フレームの位置をずらしていない場合には、ステップS174に進み、メインCPU31は、現フレームの前フレームに対する相対的位置を1画素分だけずらす処理を実行する。そして、ステップS172に戻り、再び前フレームと現フレームのブロック毎の画像間距離を同時に求める処理が行われる。

【0147】

以上のような処理がステップS173において、現フレームを全探索範囲内においてずらしたと判定されるまで繰り返し実行される。ステップS172の1回の処理により、ブロックの数の分だけ画像間距離が求められる。従って、ステップS173において、現フレームを全探索範囲内にずらしたと判定されたとき、ブロックの数の探索範囲内の画素数を乗算した数の画像間距離が得られることになる。

【0148】

ステップS175においては、1つのブロックについて、探索範囲内の画素数分の画像間距離の中から最短のものが選択され、その最短の画像間距離に対応する現フレーム上のブロックが最短ブロックとして選択される。同様の処理が各ブロックのすべてについて行われる。そして、ステップS176においては、ステップS175で求められた最短ブロックと前フレームとの動きベクトルが各ブロック毎に求められる。

【0149】

このようにして、動きベクトルを求めることにより、図28に示した場合に較べて、より迅速に動きベクトルを求めることができる。

【0150】

また、図33のフローチャートに示すように動きベクトルを求めても、図28に示した場合より、より迅速に動きベクトルを求めることができる。

【0151】

すなわち、図33の例においては、ステップS181において、ビデオカメラ35より出力された前フレームと現フレームの画像が画像メモリ43に記憶される。次に、ステップS182において、メインCPU31は、前フレームと現フレームを、それぞれ図34に示すようにピラミッドフィルタ処理することで、低解像度の画像データを作成する。このピラミッドフィルタ処理は、図19のフローチャートを参照して説明したように実行される。

【0152】

次に、ステップS183において、メインCPU31は、ステップS182で求めた低解像度の画像データを利用して、低精度の動きベクトルを求める処理を実行する。この動きベクトルを求める処理は、図28または図31を参照して説明したように実行される。ステップS183で低精度の動きベクトルが求められたとき、ステップS184に進み、メインCPU31は、元の解像度の（ピラミッドフィルタ処理をする前の）画像データを元に、各ブロック毎に低精度の動きベクトルがカバーする範囲を探索し、高精度の動きベクトルを求める処理を実行する。この動きベクトルを求める処理も、図28または図31に示したように実行される。

【0153】

次に、ハフ変換について説明する。ハフ変換は、直線を点に変換する処理であり、次のように規定される。

$$Ah(\rho, \theta) = \iint A(x, y) \delta(\rho - x \cos \theta - y \sin \theta) dx dy \quad (14)$$

上記式を変形して、

$$A h(\rho, \theta) = \int A(\rho \cos \theta - t \sin \theta, \rho \sin \theta - t \cos \theta) dt \quad (15)$$

が得られる。

【0154】

図35は、レンダリングエンジン41が、このハフ変換処理を行う場合の処理例を表している。最初に、ステップS191において、 θ に0が初期設定される。この θ は、図36に示すように、入力画像の回転角度を表している。ステップS192においては、入力画像を角度 θ だけアフィン変換した画像データが演算される。この角度 θ は、上記した(11)式における係数a乃至dを規定するものである。

【0155】

次に、ステップS193に進み、y方向にバイリニア縮小する処理が実行される。このy方向のバイリニア縮小処理の詳細については、図37のフローチャートを参照して後述するが、この処理により、図36に模式的に示すように、 θ だけ回転された入力画像が、y軸方向の1本の直線として表現されたものとなる。

【0156】

次に、ステップS194に進み、ステップS193の処理で直線状に縮小された点列を、描画エリア52上の角度 θ の位置に書き込む処理が実行される。

【0157】

次に、ステップS195に進み、 θ が π に等しいか否かが判定され、等しくない場合には、ステップS196に進み、 θ が π/n だけインクリメントされ、ステップS192に戻り、それ以降の処理が繰り返し実行される。ステップS195において、 θ が π と等しいと判定された場合、処理が終了される。

【0158】

これにより、例えば 64×64 画素の画像を、角度分解能を π/n とするとき、最大でも $6n$ 個のポリゴンで、ハフ変換することができる。また、 256×256 画素の画像データの場合、角度分解能を π/n とするとき、最大でも $8n$ 個のポリゴンでハフ変換することができる。

【0159】

次に、図35のステップS193におけるy方向にバイリニア縮小する処理の詳細について説明する。この処理をレンダリングエンジン41に実行させる場合、メインCPU31は、例えば次のようなコマンドをレンダリングエンジン41に供給する。

```
int L; /*ソースエリアの一辺の長さ*/
int offset;
L0=2*N; /*初期画像の一辺の長さ*/
L=L0;
offset=0;
while(L>1) {
    Set_Texture_Base(0,offset); /*テクスチャエリアのベースポイント設定*/
    offset +=L;
    Set_Drawing_Base(0,offset); /*描画エリアのベースポイント設定*/
    Flat_Texture_Rectangle(0,0, L0,0, L0,L/2, 0,L/2,
                           0,0.5, L0,0.5, L0,L+0.5, 0,L+0.5,1.0);
    L=L/2;
}
```

【0160】

この処理をフローチャートに表すと、図37に示すようになる。最初に、ステップS201において、変数Lに L_0 が初期設定されるとともに、変数offsetに0が初期設定される。次に、ステップS202において、テクスチャエリア51のベースポイントを(0, offset)に設定する処理が実行される。すなわち、図38に示すように、ベースポイントT(0, 0)が設定される。次に、ステップS203に進み、変数offsetがLだけインクリメントされる。そして、ステップS204において、描画エリア52のベースポイントに(0, offset)が設定される。いまの場合、図38に示すように、ベースポイントD(0, L)が設定される。

【0161】

次に、ステップS205において、ソース（テクチャエリア）の4角形（0, 0.5, L_0 , 0.5, L_0 , $L+0.5$, 0, $L+0.5$ ）の画素値に1を乗算して、デスティネーションの4角形（0, 0, L_0 , 0, L_0 , $L/2$, 0, $L/2$ ）に加算して描画する処理が実行される。

【0162】

次に、ステップS206に進み、変数Lが現在の値の $1/2$ とされる。ステップS207において、変数Lが1より大きいかが判定され、変数Lが1より大きい場合には、ステップS202に戻り、それ以降の処理が繰り返し実行される。

【0163】

以下、同様の処理が繰り返し実行され、ステップS207において、変数Lが1より大きくないと判定された場合（変数Lが1に等しいと判定された場合）、y方向のバイリニア縮小処理が終了される。

【0164】

次に、モーションブラー処理について説明する。図39に示すように、直前のフレーム F_1 と現在のフレーム F_2 の間に動きがある場合、その間に動きベクトルが存在する。電子シャッター付きのCCDで撮像すると、高速に移動する物体でもブレなく撮像することができるが、その動きが早い場合（動きベクトルの値が大きい場合）、フレーム F_1 の次にフレーム F_2 をそのまま表示すると、画像のブレが存在しないので、画像が滑らかに動いて見えないときがある。このような場合、動きベクトルに沿って、n個のコンボリューション係数（コンボリューション係数Cの値は、 $1.0/n$ とされる）を設定し、画像をn回重ね書きした画像を生成する。例えば図39の例においては、 $n=5$ とされているので、フレーム F_{11} 乃至フレーム F_{15} を重ね書きした画像を直前のフレーム F_1 の次に、現フレーム F_2 に代えて出力する。これにより、ユーザに、滑らかに移動している画像として画像を認識させることが可能となる。これが、モーションブラー処理である。

【0165】

次に、図40のフローチャートを参照して、レンダリングエンジン41が行う

モーションブラー処理について説明する。最初に、ステップ S 2 2 1 において、直前のフレームと現フレームから動きベクトルを検出する処理が実行される。すなわち、例えば、図 4 1 に示すように、電子シャッタ付きの CCD を内蔵するビデオカメラ 3 5 より出力された画像のうち、現在のフレームとその直前のフレームの画像の動きベクトルを抽出する処理が実行される。次に、ステップ S 2 2 2 において、動きベクトルに沿った n 個のコンボルーション係数 C が決定される。上述したように、このコンボルーション係数 C の値は、補間するフレームの数を n とするとき、 $1/n$ とされる。

【0166】

次に、ステップ S 2 2 3 において、ブレンディングモードとして、モード 1 が設定される。ステップ S 2 2 4 においては、コンボルーション係数の変更回数を示す変数 N が 0 に初期設定される。次に、ステップ S 2 2 5 において、直前のフレームの画像がソース画像とされる。ステップ S 2 2 6 においては、ソースに 1 個のコンボルーション係数を乗算し、デスティネーションに加算して描画する処理が実行される。

【0167】

次に、ステップ S 2 2 7 に進み、 n 個のコンボルーション係数をすべて乗算したか否かが判定され、まだ乗算していないコンボルーション係数が存在する場合には、ステップ S 2 2 8 に進み、コンボルーション係数を変更する処理が実行される。そして、ステップ S 2 2 9 において、コンボルーション係数の変更回数を示す変数 N が 1 だけインクリメントされる。

【0168】

ステップ S 2 3 0 においては、コンボルーション係数の変更回数 N が $n/2$ より大きいか否かが判定される。変数 N が $n/2$ より小さい場合、ステップ S 2 2 5 に戻り、それ以降の処理が繰り返し実行される。

【0169】

ステップ S 2 3 0 において、変数 N が $n/2$ より大きいと判定された場合、ステップ S 2 3 1 に進み、現フレームの画像がソースフレームとされる。そして、ステップ S 2 2 6 に戻り、それ以降の処理が繰り返し実行される。

【0170】

ステップS227において、 n 個のコンボルーション係数をすべて乗算したと判定された場合、処理が終了される。

【0171】

すなわち、図42に示すように、コンボルーションフィルタ係数が、4個($n=4$)の場合、直前のフレーム F_1 に近いフレーム F_{21} 、 F_{22} (第1回目と第2回目の描画より生成されるフレーム)は、直前のフレーム F_1 を元に生成され、フレーム F_{23} 、 F_{24} (第3回目と第4回目の描画より生成されるフレーム)は、現フレーム F_2 に近いので、現フレーム F_2 を元に生成される。この場合、動きベクトルの方向が本来の方向と逆方向になるので、そのことを考慮して描画処理が実行される。

【0172】

なお、動きベクトルは、図41に示すように、マクロブロック単位で存在するため、このモーションブラー処理は、マクロブロック単位で実行される。

【0173】

レンダリングエンジン41に、以上のような各種の処理を実行させたとき、その処理の結果をCRT36に、例えば図43に示すように表示させることができる。この場合、メインCPU31は、図44のフローチャートに示すような処理を実行する。

【0174】

すなわち、最初にステップS251において、メインCPU31は、ビデオカメラ35より入力されたビデオ画像を、PCRTC44とメモリアンターフェース42を介して取り込ませ、画像メモリ43に記憶させる。

【0175】

次に、ステップS252において、メインCPU31は、レンダリングエンジン41を制御し、図22のフローチャートを参照して説明したように、フレーム間差分処理(動き抽出処理)を実行させる。

【0176】

次に、ステップS253において、メインCPU31は、レンダリングエンジン

41を制御し、エッジ抽出処理を実行させる。このエッジ抽出処理は、図7を参照して説明したコンボリューションフィルタ処理のコンボリューションフィルタ係数として、図45に示すような値を設定することで行われる。

【0177】

次に、ステップS254に進み、メインCPU31は、レンダリングエンジン41を制御し、図35のフローチャートを参照して説明したようなハフ変換処理を実行させる。さらに、ステップS255において、メインCPU31は、レンダリングエンジン41を制御し、ステップS254で行われたハフ変換処理を元に、線分抽出処理を実行させる。

【0178】

ステップS256においては、ステップS252乃至ステップS255の処理結果の表示がユーザにより指令されているか否かが判定され、指令されていない場合には、ステップS251に戻り、それ以降の処理が繰り返し実行される。ステップS256において、処理結果の表示が指令されていると判定された場合、ステップS257に進み、メインCPU31は、PCRTC44を制御し、画像メモリ43に描画されている画像をメモリインターフェース46を介して読み出し、CRT36に出力し、表示させる。これにより、例えば図43に示すような画像がCRT36に表示される。

【0179】

図43の表示例においては、その最も左側に、ステップS252におけるフレーム間差分処理（動き抽出処理）の画像が表示されており、その右側に、ステップS253におけるエッジ抽出処理の画像が表示されている。

【0180】

さらに、その右側には、最も下側に、ステップS251で取り込まれた入力画像が表示され、その上方には、ステップS254におけるハフ変換処理の画像（図36における $A_h(\rho, \theta)$ の出力画像）が表示され、さらにその上側には、ステップS255における線分抽出処理の結果が（ハフ変換により抽出された線分が）表示されている。

【0181】

以上においては、本発明をコンピュータエンタテインメント装置に応用した場合を例として説明したが、本発明は、その他の画像処理装置に応用することも可能である。

【0182】

なお、本明細書において、システムとは、複数の装置により構成される装置全体を表すものとする。

【0183】

なお、上記したような処理を行うコンピュータプログラムをユーザに提供する提供媒体としては、磁気ディスク、CD-ROM、固体メモリなどの記録媒体の他、ネットワーク、衛星などの通信媒体を利用することができる。

【0184】

【発明の効果】

以上の如く、請求項1に記載の画像処理装置、請求項8に記載の画像処理方法、および請求項9に記載の提供媒体によれば、画素単位で所定の演算を施して第2の記憶手段にポリゴン単位で描画する動作を、所定の演算結果が得られるまで繰り返し実行させるようにしたので、簡単な構成で、かつ、低コストの装置で、画像処理を実行することが可能となる。

【0185】

請求項10に記載の画像処理装置、請求項17に記載の画像処理方法、および請求項18に記載の提供媒体によれば、画素単位で所定の演算を施してポリゴン単位で描画する動作を、所定の演算結果が得られるまで繰り返し行わせる描画コマンドを生成するようにしたので、描画コマンドを用いて、所定の演算を実行させることが可能となる。

【0186】

請求項19に記載の画像処理装置、請求項26に記載の画像処理方法、および請求項27に記載の提供媒体によれば、画素単位で所定の演算のうちの一部の演算を施して第2の記憶手段にポリゴン単位で描画するとともに、さらに、画素単位で所定の演算のうちの一部の演算を施して、画像データに加算または減算

して、第2の記憶手段にポリゴン単位で描画するようにしたので、描画処理により、各種の演算処理を迅速に実行させることが可能となる。

【図面の簡単な説明】

【図1】

本発明の画像処理装置の構成例を示すブロック図である。

【図2】

図1の画像メモリ43の内部の記録エリアを説明する図である。

【図3】

バイリニア補間処理を説明する図である。

【図4】

コンボルーションフィルタ処理を説明する図である。

【図5】

ソースの画素データを説明する図である。

【図6】

コンボルーションフィルタ係数を説明する図である。

【図7】

コンボルーションフィルタ処理を説明するフローチャートである。

【図8】

デスティネーションの画素データを説明する図である。

【図9】

デスティネーションの画素データを説明する図である。

【図10】

デスティネーションの画素データを説明する図である。

【図11】

デスティネーションの画素データを説明する図である。

【図12】

デスティネーションの画素データを説明する図である。

【図13】

デスティネーションの画素データを説明する図である。

【図 14】

コンボルーションフィルタ処理の他の例を説明するフローチャートである。

【図 15】

コンボルーションフィルタ処理の他の例を説明するフローチャートである。

【図 16】

図 14 のステップ S 5 3 におけるデスティネーションの画素データを説明する図である。

【図 17】

コンボルーションフィルタ処理のさらに他の例を説明するフローチャートである。

【図 18】

ピラミッドフィルタ処理を説明する図である。

【図 19】

ピラミッドフィルタ処理を説明するフローチャートである。

【図 20】

図 19 におけるステップ S 9 2, S 9 4 の処理を説明する図である。

【図 21】

フレーム間差分処理を説明する図である。

【図 22】

フレーム間差分処理を説明するフローチャートである。

【図 23】

画像間距離を説明する図である。

【図 24】

画像間距離演算処理を説明するフローチャートである。

【図 25】

パターンマッチング処理を説明するフローチャートである。

【図 26】

パターンマッチング処理の他の例を説明するフローチャートである。

【図27】

パターンマッチング処理を説明する図である。

【図28】

動きベクトル検出処理を説明するフローチャートである。

【図29】

動きベクトルを抽出する処理を説明する図である。

【図30】

ブロックを探索範囲内でずらして動きベクトルを求める処理を説明する図である。

【図31】

他の動きベクトル検出処理を説明するフローチャートである。

【図32】

図31のステップS172における画像間距離を求める処理を説明する図である。

【図33】

さらに他の動きベクトル検出処理を説明するフローチャートである。

【図34】

図33のステップS182におけるピラミッドフィルタ処理を説明する図である。

【図35】

ハフ変換処理を説明するフローチャートである。

【図36】

ハフ変換を説明する図である。

【図37】

図35のステップS193におけるy方向のバイリニア縮小処理を説明するフローチャートである。

【図38】

図37のステップS202, S204の処理を説明する図である。

【図 39】

モーションブラー処理を説明する図である。

【図 40】

モーションブラー処理を説明するフローチャートである。

【図 41】

電子シャッター付きCCD動画像の補正処理を説明する図である。

【図 42】

モーションブラー処理を説明する図である。

【図 43】

画像表示の例を示す図である。

【図 44】

画像表示処理を説明するフローチャートである。

【図 45】

エッジ抽出処理を行う場合のフィルタ係数の例を示す図である。

【図 46】

従来の画像処理装置の構成例を示すブロック図である。

【図 47】

従来のコンボルーションフィルタ処理を説明するフローチャートである。

【図 48】

ソースの画素データを説明する図である。

【図 49】

コンボルーションフィルタ係数を説明する図である。

【図 50】

デスティネーションの画素データを説明する図である。

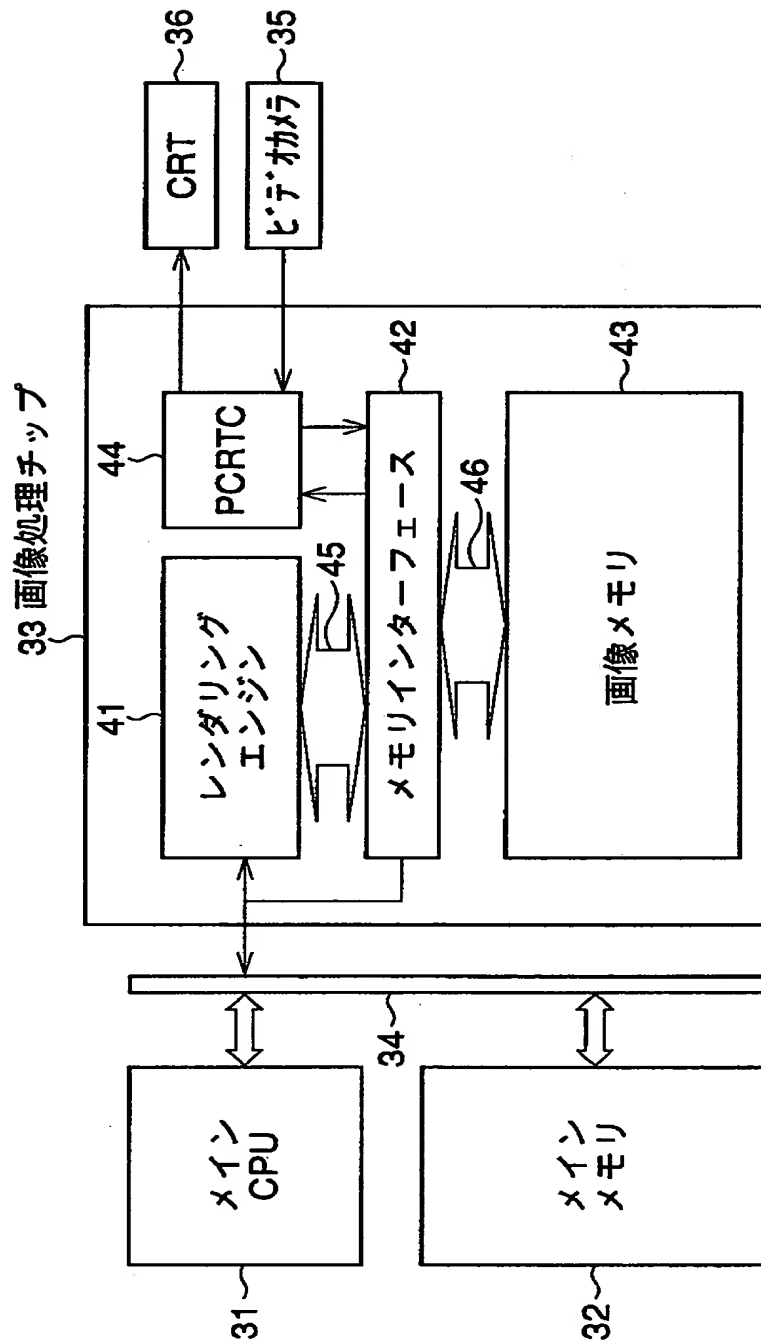
【符号の説明】

31 メインCPU, 32 メインメモリ, 33 画像処理チップ, 34
バス, 35 ビデオカメラ, 36 CRT, 41 レンダリングエンジン
, 42 メモリインターフェース, 43 画像メモリ, 44 プログラマ
ブルCRTコントローラ, 45, 46 バス, 51 テクスチャエリア, 5

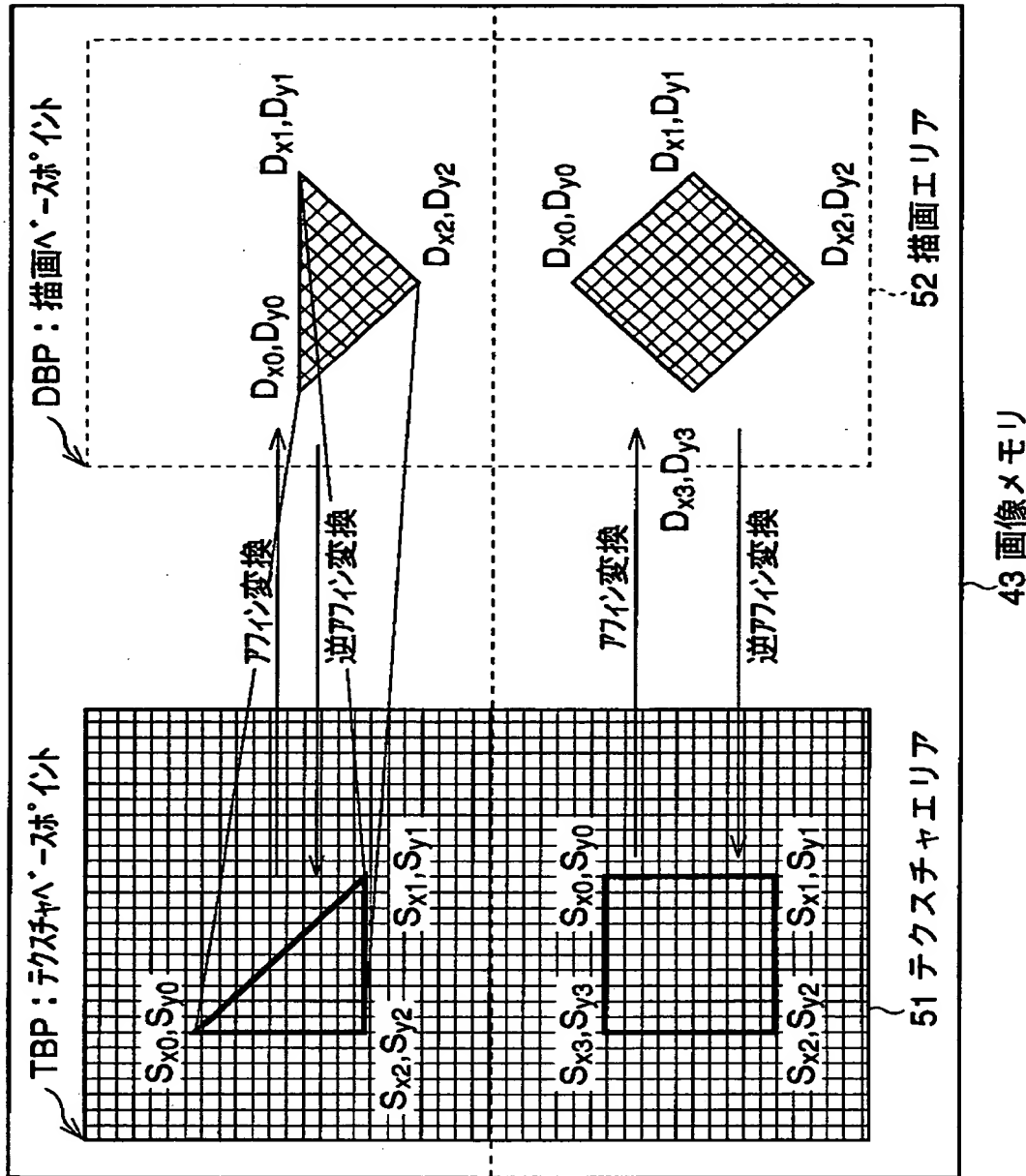
2 描画エリア

【書類名】 図面

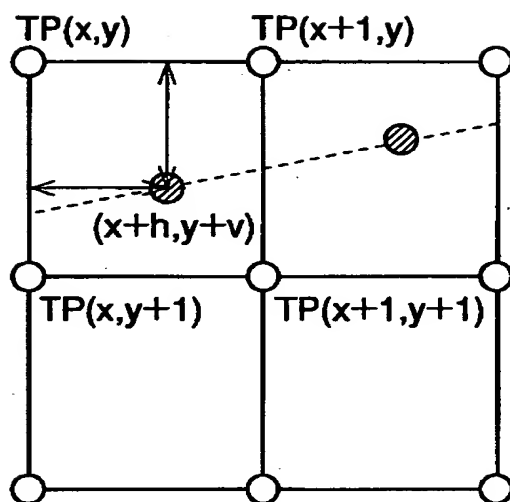
【図 1】



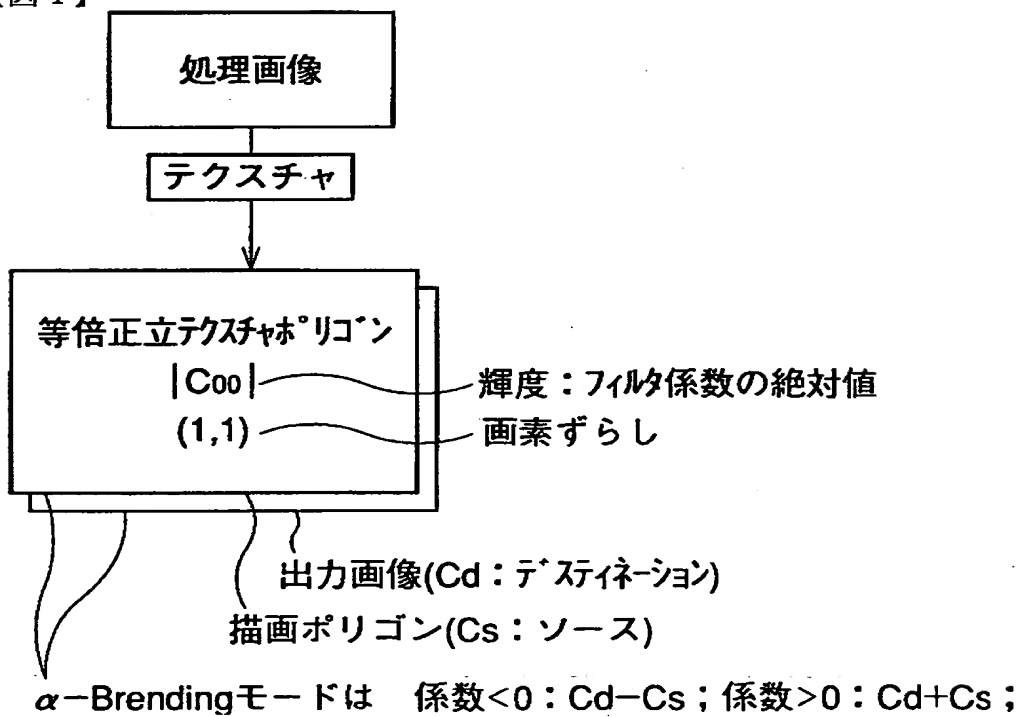
【図 2】



【図 3】



【図 4】



【図 5】

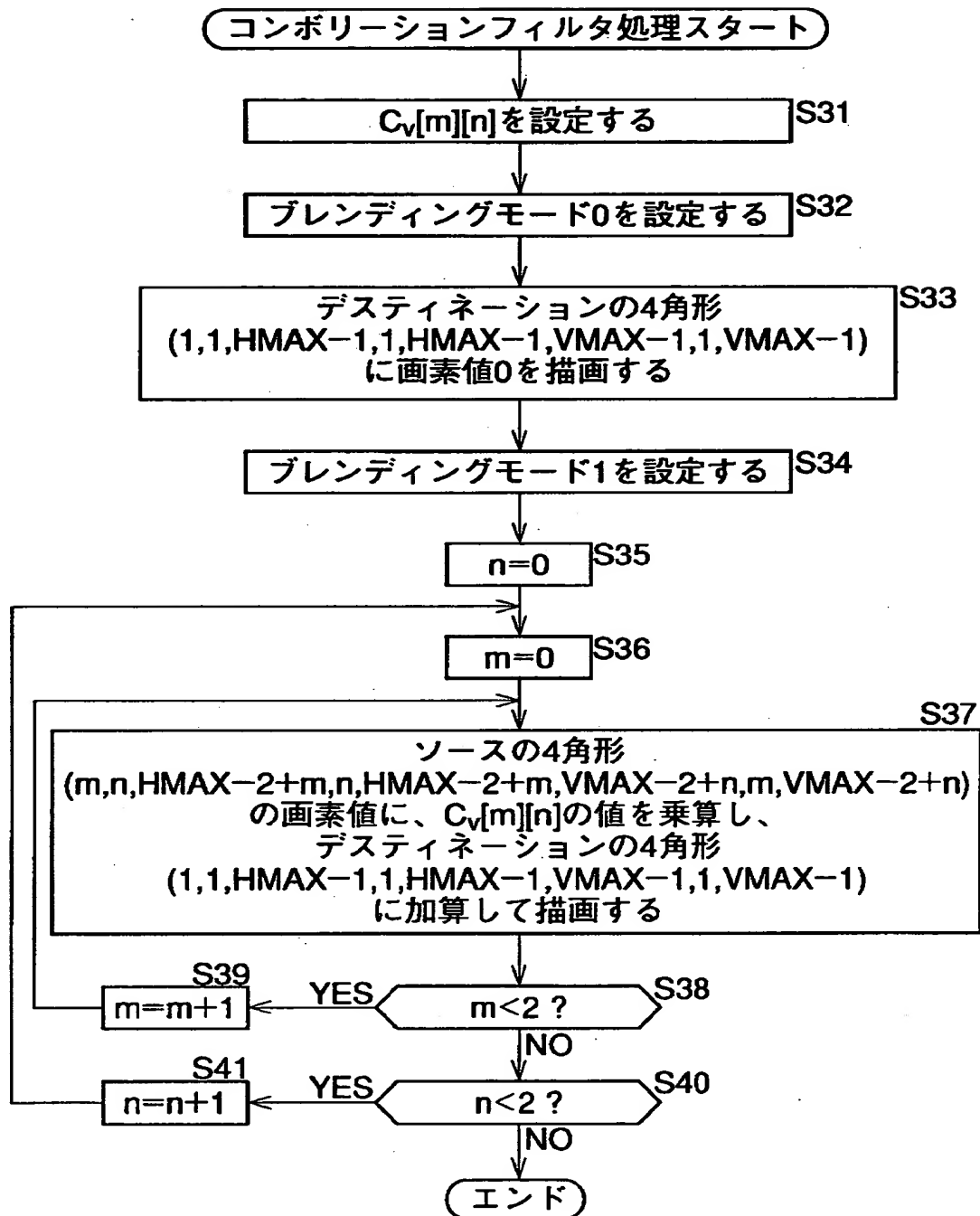
		C_{sp}					
		0	1	2	3	4	5 (HMAX-1)
j	0						
	1						
	2						
	3						
	4						
	5 (VMAX-1)						

【図 6】

		m		
		0	1	2
n	0	C ₀₀ (1,1)	C ₀₁ (0,1)	C ₀₂ (-1,1)
	1	C ₁₀ (1,0)	C ₁₁ (0,0)	C ₁₂ (-1,0)
	2	C ₂₀ (1,-1)	C ₂₁ (0,-1)	C ₂₂ (-1,-1)

コンボリューションフィルタ係数

【図 7】



【図 8】

		C_{dp}					
		0	1	2	3	4	5 (HMAX-1)
j	0						
	1		0	0	0	0	0
	2		0	0	0	0	0
	3		0	0	0	0	0
	4		0	0	0	0	0
	5 (VMAX-1)		0	0	0	0	0

【図 9】

		C_{dp}					
		0	1	2	3	4	5 (HMAX-1)
j	0						
	1		$C_{sp00} \times C_{00}$	$C_{sp10} \times C_{00}$	$C_{sp20} \times C_{00}$	$C_{sp30} \times C_{00}$	$C_{sp40} \times C_{00}$
	2		$C_{sp01} \times C_{00}$	$C_{sp11} \times C_{00}$	$C_{sp21} \times C_{00}$	$C_{sp31} \times C_{00}$	$C_{sp41} \times C_{00}$
	3		$C_{sp02} \times C_{00}$	$C_{sp12} \times C_{00}$	$C_{sp22} \times C_{00}$	$C_{sp32} \times C_{00}$	$C_{sp42} \times C_{00}$
	4		$C_{sp03} \times C_{00}$	$C_{sp13} \times C_{00}$	$C_{sp23} \times C_{00}$	$C_{sp33} \times C_{00}$	$C_{sp43} \times C_{00}$
	5 (VMAX-1)		$C_{sp04} \times C_{00}$	$C_{sp14} \times C_{00}$	$C_{sp24} \times C_{00}$	$C_{sp34} \times C_{00}$	$C_{sp44} \times C_{00}$

【図10】

		C_{dp}					
		0	1	2	3	4	5 (HMAX-1)
j	0						
	1		$C_{sp00} \times C_{00} + C_{sp10} \times C_{10}$	$C_{sp10} \times C_{00} + C_{sp20} \times C_{10}$	$C_{sp20} \times C_{00} + C_{sp30} \times C_{10}$	$C_{sp30} \times C_{00} + C_{sp40} \times C_{10}$	$C_{sp40} \times C_{00} + C_{sp50} \times C_{10}$
	2		$C_{sp01} \times C_{00} + C_{sp11} \times C_{10}$	$C_{sp11} \times C_{00} + C_{sp21} \times C_{10}$	$C_{sp21} \times C_{00} + C_{sp31} \times C_{10}$	$C_{sp31} \times C_{00} + C_{sp41} \times C_{10}$	$C_{sp41} \times C_{00} + C_{sp51} \times C_{10}$
	3		$C_{sp02} \times C_{00} + C_{sp12} \times C_{10}$	$C_{sp12} \times C_{00} + C_{sp22} \times C_{10}$	$C_{sp22} \times C_{00} + C_{sp32} \times C_{10}$	$C_{sp32} \times C_{00} + C_{sp42} \times C_{10}$	$C_{sp42} \times C_{00} + C_{sp52} \times C_{10}$
	4		$C_{sp03} \times C_{00} + C_{sp13} \times C_{10}$	$C_{sp13} \times C_{00} + C_{sp23} \times C_{10}$	$C_{sp23} \times C_{00} + C_{sp33} \times C_{10}$	$C_{sp33} \times C_{00} + C_{sp43} \times C_{10}$	$C_{sp43} \times C_{00} + C_{sp53} \times C_{10}$
	5 (VMAX-1)		$C_{sp04} \times C_{00} + C_{sp14} \times C_{10}$	$C_{sp14} \times C_{00} + C_{sp24} \times C_{10}$	$C_{sp24} \times C_{00} + C_{sp34} \times C_{10}$	$C_{sp34} \times C_{00} + C_{sp44} \times C_{10}$	$C_{sp44} \times C_{00} + C_{sp54} \times C_{10}$

【図11】

		C_{dp}					
		0	1	2	3	4	5 (HMAX-1)
j	0						
	1		$C_{sp00} \times C_{00}$ $+ C_{sp10} \times C_{10}$ $+ C_{sp20} \times C_{20}$	$C_{sp10} \times C_{00}$ $+ C_{sp20} \times C_{10}$ $+ C_{sp30} \times C_{20}$	$C_{sp20} \times C_{00}$ $+ C_{sp30} \times C_{10}$ $+ C_{sp40} \times C_{20}$	$C_{sp30} \times C_{00}$ $+ C_{sp40} \times C_{10}$ $+ C_{sp50} \times C_{20}$	$C_{sp40} \times C_{00}$ $+ C_{sp50} \times C_{10}$
	2		$C_{sp01} \times C_{00}$ $+ C_{sp11} \times C_{10}$ $+ C_{sp21} \times C_{20}$	$C_{sp11} \times C_{00}$ $+ C_{sp21} \times C_{10}$ $+ C_{sp31} \times C_{20}$	$C_{sp21} \times C_{00}$ $+ C_{sp31} \times C_{10}$ $+ C_{sp41} \times C_{20}$	$C_{sp31} \times C_{00}$ $+ C_{sp41} \times C_{10}$ $+ C_{sp51} \times C_{20}$	$C_{sp41} \times C_{00}$ $+ C_{sp51} \times C_{10}$
	3		$C_{sp02} \times C_{00}$ $+ C_{sp12} \times C_{10}$ $+ C_{sp22} \times C_{20}$	$C_{sp12} \times C_{00}$ $+ C_{sp22} \times C_{10}$ $+ C_{sp32} \times C_{20}$	$C_{sp22} \times C_{00}$ $+ C_{sp32} \times C_{10}$ $+ C_{sp42} \times C_{20}$	$C_{sp32} \times C_{00}$ $+ C_{sp42} \times C_{10}$ $+ C_{sp52} \times C_{20}$	$C_{sp42} \times C_{00}$ $+ C_{sp52} \times C_{10}$
	4		$C_{sp03} \times C_{00}$ $+ C_{sp13} \times C_{10}$ $+ C_{sp23} \times C_{20}$	$C_{sp13} \times C_{00}$ $+ C_{sp23} \times C_{10}$ $+ C_{sp33} \times C_{20}$	$C_{sp23} \times C_{00}$ $+ C_{sp33} \times C_{10}$ $+ C_{sp43} \times C_{20}$	$C_{sp33} \times C_{00}$ $+ C_{sp43} \times C_{10}$ $+ C_{sp53} \times C_{20}$	$C_{sp43} \times C_{00}$ $+ C_{sp53} \times C_{10}$
	5 (VMAX-1)		$C_{sp04} \times C_{00}$ $+ C_{sp14} \times C_{10}$ $+ C_{sp24} \times C_{20}$	$C_{sp14} \times C_{00}$ $+ C_{sp24} \times C_{10}$ $+ C_{sp34} \times C_{20}$	$C_{sp24} \times C_{00}$ $+ C_{sp34} \times C_{10}$ $+ C_{sp44} \times C_{20}$	$C_{sp34} \times C_{00}$ $+ C_{sp44} \times C_{10}$ $+ C_{sp54} \times C_{20}$	$C_{sp44} \times C_{00}$ $+ C_{sp54} \times C_{10}$

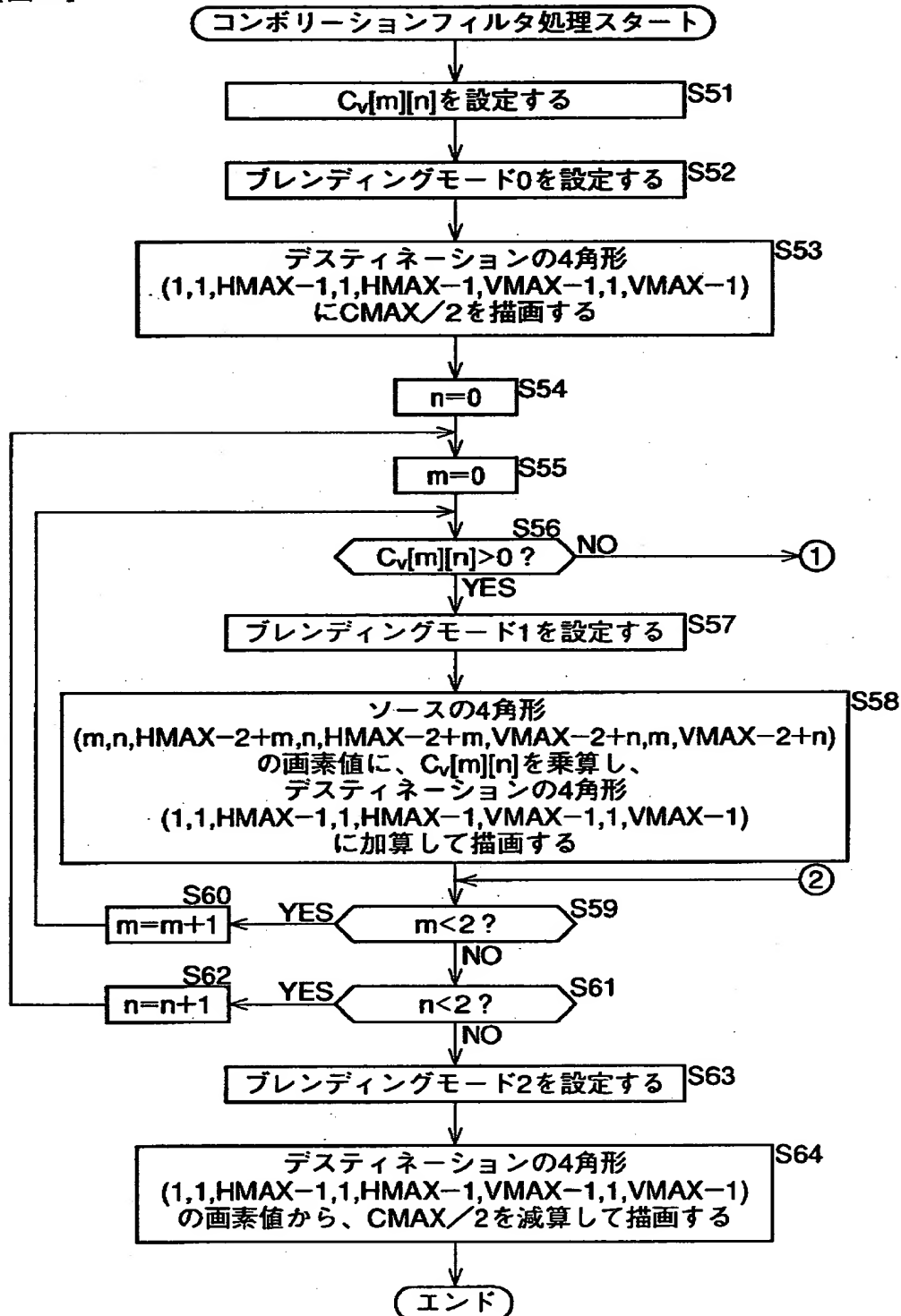
【図12】

		C_{dp}					
		0	1	2	3	4	5 (HMAX-1)
j	0						
	1		$C_{sp00} \times C_{00}$ $+C_{sp10} \times C_{10}$ $+C_{sp20} \times C_{20}$ $+C_{sp01} \times C_{01}$	$C_{sp10} \times C_{00}$ $+C_{sp20} \times C_{10}$ $+C_{sp30} \times C_{20}$ $+C_{sp11} \times C_{01}$	$C_{sp20} \times C_{00}$ $+C_{sp30} \times C_{10}$ $+C_{sp40} \times C_{20}$ $+C_{sp21} \times C_{01}$	$C_{sp30} \times C_{00}$ $+C_{sp40} \times C_{10}$ $+C_{sp50} \times C_{20}$ $+C_{sp31} \times C_{01}$	$C_{sp40} \times C_{00}$ $+C_{sp50} \times C_{10}$ $+C_{sp41} \times C_{01}$
	2		$C_{sp01} \times C_{00}$ $+C_{sp11} \times C_{10}$ $+C_{sp21} \times C_{20}$ $+C_{sp02} \times C_{01}$	$C_{sp11} \times C_{00}$ $+C_{sp21} \times C_{10}$ $+C_{sp31} \times C_{20}$ $+C_{sp12} \times C_{01}$	$C_{sp21} \times C_{00}$ $+C_{sp31} \times C_{10}$ $+C_{sp41} \times C_{20}$ $+C_{sp22} \times C_{01}$	$C_{sp31} \times C_{00}$ $+C_{sp41} \times C_{10}$ $+C_{sp51} \times C_{20}$ $+C_{sp32} \times C_{01}$	$C_{sp41} \times C_{00}$ $+C_{sp51} \times C_{10}$ $+C_{sp42} \times C_{01}$
	3		$C_{sp02} \times C_{00}$ $+C_{sp12} \times C_{10}$ $+C_{sp22} \times C_{20}$ $+C_{sp03} \times C_{01}$	$C_{sp12} \times C_{00}$ $+C_{sp22} \times C_{10}$ $+C_{sp32} \times C_{20}$ $+C_{sp13} \times C_{01}$	$C_{sp22} \times C_{00}$ $+C_{sp32} \times C_{10}$ $+C_{sp42} \times C_{20}$ $+C_{sp23} \times C_{01}$	$C_{sp32} \times C_{00}$ $+C_{sp42} \times C_{10}$ $+C_{sp52} \times C_{20}$ $+C_{sp33} \times C_{01}$	$C_{sp42} \times C_{00}$ $+C_{sp52} \times C_{10}$ $+C_{sp43} \times C_{01}$
	4		$C_{sp03} \times C_{00}$ $+C_{sp13} \times C_{10}$ $+C_{sp23} \times C_{20}$ $+C_{sp04} \times C_{01}$	$C_{sp13} \times C_{00}$ $+C_{sp23} \times C_{10}$ $+C_{sp33} \times C_{20}$ $+C_{sp14} \times C_{01}$	$C_{sp23} \times C_{00}$ $+C_{sp33} \times C_{10}$ $+C_{sp43} \times C_{20}$ $+C_{sp24} \times C_{01}$	$C_{sp33} \times C_{00}$ $+C_{sp43} \times C_{10}$ $+C_{sp53} \times C_{20}$ $+C_{sp34} \times C_{01}$	$C_{sp43} \times C_{00}$ $+C_{sp53} \times C_{10}$ $+C_{sp44} \times C_{01}$
	5 (VMAX-1)		$C_{sp04} \times C_{00}$ $+C_{sp14} \times C_{10}$ $+C_{sp24} \times C_{20}$ $+C_{sp05} \times C_{01}$	$C_{sp14} \times C_{00}$ $+C_{sp24} \times C_{10}$ $+C_{sp34} \times C_{20}$ $+C_{sp15} \times C_{01}$	$C_{sp24} \times C_{00}$ $+C_{sp34} \times C_{10}$ $+C_{sp44} \times C_{20}$ $+C_{sp25} \times C_{01}$	$C_{sp34} \times C_{00}$ $+C_{sp44} \times C_{10}$ $+C_{sp54} \times C_{20}$ $+C_{sp35} \times C_{01}$	$C_{sp44} \times C_{00}$ $+C_{sp54} \times C_{10}$ $+C_{sp45} \times C_{01}$

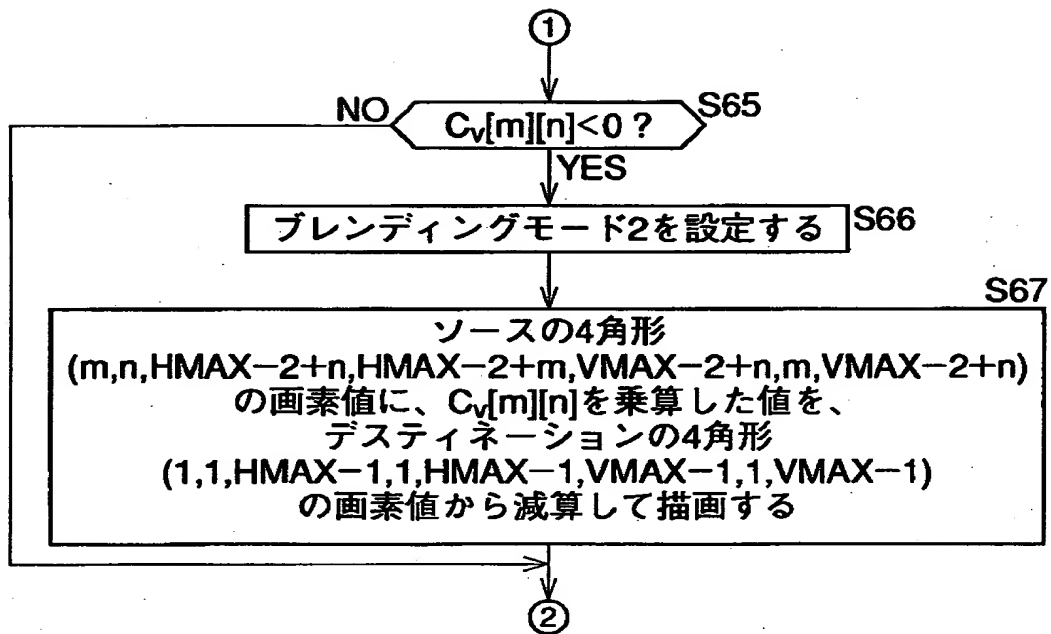
【図13】

		C_{dp}					
		0	1	2	3	4	5 (HMAX-1)
j	0						
	1		$C_{sp00} \times C_{00}$ $+C_{sp10} \times C_{10}$ $+C_{sp20} \times C_{20}$ $+C_{sp01} \times C_{01}$ \vdots $+C_{sp22} \times C_{22}$	$C_{sp10} \times C_{00}$ $+C_{sp20} \times C_{10}$ $+C_{sp30} \times C_{20}$ $+C_{sp11} \times C_{01}$ \vdots $+C_{sp32} \times C_{22}$	$C_{sp20} \times C_{00}$ $+C_{sp30} \times C_{10}$ $+C_{sp40} \times C_{20}$ $+C_{sp21} \times C_{01}$ \vdots $+C_{sp42} \times C_{22}$	$C_{sp30} \times C_{00}$ $+C_{sp40} \times C_{10}$ $+C_{sp50} \times C_{20}$ $+C_{sp31} \times C_{01}$ \vdots $+C_{sp52} \times C_{22}$	$C_{sp40} \times C_{00}$ $+C_{sp50} \times C_{10}$ $+C_{sp41} \times C_{01}$ \vdots
	2		$C_{sp01} \times C_{00}$ $+C_{sp11} \times C_{10}$ $+C_{sp21} \times C_{20}$ $+C_{sp02} \times C_{01}$ \vdots $+C_{sp23} \times C_{22}$	$C_{sp11} \times C_{00}$ $+C_{sp21} \times C_{10}$ $+C_{sp31} \times C_{20}$ $+C_{sp12} \times C_{01}$ \vdots $+C_{sp33} \times C_{22}$	$C_{sp21} \times C_{00}$ $+C_{sp31} \times C_{10}$ $+C_{sp41} \times C_{20}$ $+C_{sp22} \times C_{01}$ \vdots $+C_{sp43} \times C_{22}$	$C_{sp31} \times C_{00}$ $+C_{sp41} \times C_{10}$ $+C_{sp51} \times C_{20}$ $+C_{sp32} \times C_{01}$ \vdots $+C_{sp53} \times C_{22}$	$C_{sp41} \times C_{00}$ $+C_{sp51} \times C_{10}$ $+C_{sp42} \times C_{01}$ \vdots
	3		$C_{sp02} \times C_{00}$ $+C_{sp12} \times C_{10}$ $+C_{sp22} \times C_{20}$ $+C_{sp03} \times C_{01}$ \vdots $+C_{sp24} \times C_{22}$	$C_{sp12} \times C_{00}$ $+C_{sp22} \times C_{10}$ $+C_{sp32} \times C_{20}$ $+C_{sp13} \times C_{01}$ \vdots $+C_{sp34} \times C_{22}$	$C_{sp22} \times C_{00}$ $+C_{sp32} \times C_{10}$ $+C_{sp42} \times C_{20}$ $+C_{sp23} \times C_{01}$ \vdots $+C_{sp44} \times C_{22}$	$C_{sp32} \times C_{00}$ $+C_{sp42} \times C_{10}$ $+C_{sp52} \times C_{20}$ $+C_{sp33} \times C_{01}$ \vdots $+C_{sp54} \times C_{22}$	$C_{sp42} \times C_{00}$ $+C_{sp52} \times C_{10}$ $+C_{sp43} \times C_{01}$ \vdots
	4		$C_{sp03} \times C_{00}$ $+C_{sp13} \times C_{10}$ $+C_{sp23} \times C_{20}$ $+C_{sp04} \times C_{01}$ \vdots $+C_{sp25} \times C_{22}$	$C_{sp13} \times C_{00}$ $+C_{sp23} \times C_{10}$ $+C_{sp33} \times C_{20}$ $+C_{sp14} \times C_{01}$ \vdots $+C_{sp35} \times C_{22}$	$C_{sp23} \times C_{00}$ $+C_{sp33} \times C_{10}$ $+C_{sp43} \times C_{20}$ $+C_{sp24} \times C_{01}$ \vdots $+C_{sp45} \times C_{22}$	$C_{sp33} \times C_{00}$ $+C_{sp43} \times C_{10}$ $+C_{sp53} \times C_{20}$ $+C_{sp34} \times C_{01}$ \vdots $+C_{sp55} \times C_{22}$	$C_{sp43} \times C_{00}$ $+C_{sp53} \times C_{10}$ $+C_{sp44} \times C_{01}$ \vdots
	5 (VMAX-1)		$C_{sp04} \times C_{00}$ $+C_{sp14} \times C_{10}$ $+C_{sp24} \times C_{20}$ $+C_{sp05} \times C_{01}$ \vdots	$C_{sp14} \times C_{00}$ $+C_{sp24} \times C_{10}$ $+C_{sp34} \times C_{20}$ $+C_{sp15} \times C_{01}$ \vdots	$C_{sp24} \times C_{00}$ $+C_{sp34} \times C_{10}$ $+C_{sp44} \times C_{20}$ $+C_{sp25} \times C_{01}$ \vdots	$C_{sp34} \times C_{00}$ $+C_{sp44} \times C_{10}$ $+C_{sp54} \times C_{20}$ $+C_{sp35} \times C_{01}$ \vdots	$C_{sp44} \times C_{00}$ $+C_{sp54} \times C_{10}$ $+C_{sp45} \times C_{01}$ \vdots

【図14】



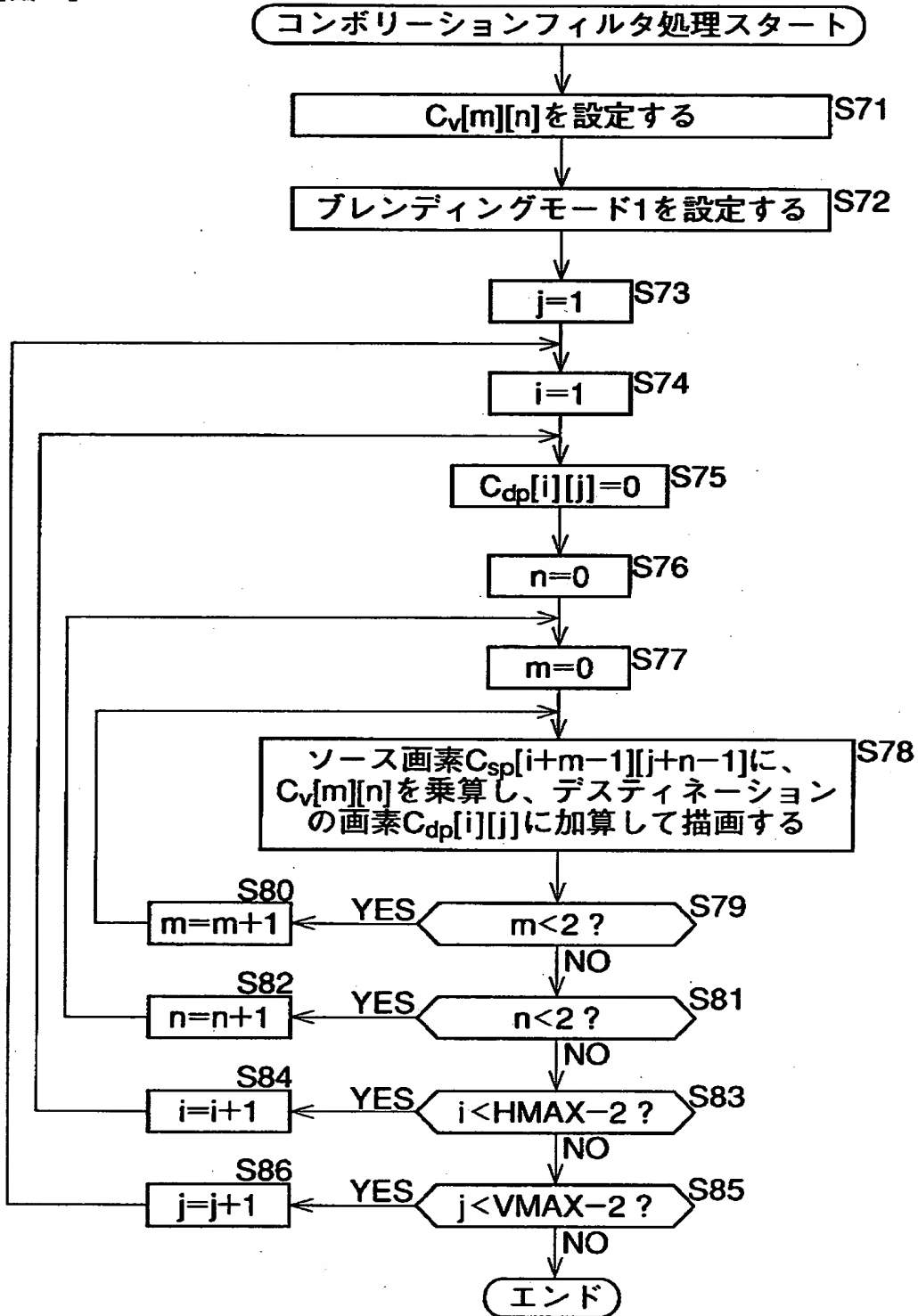
【図15】



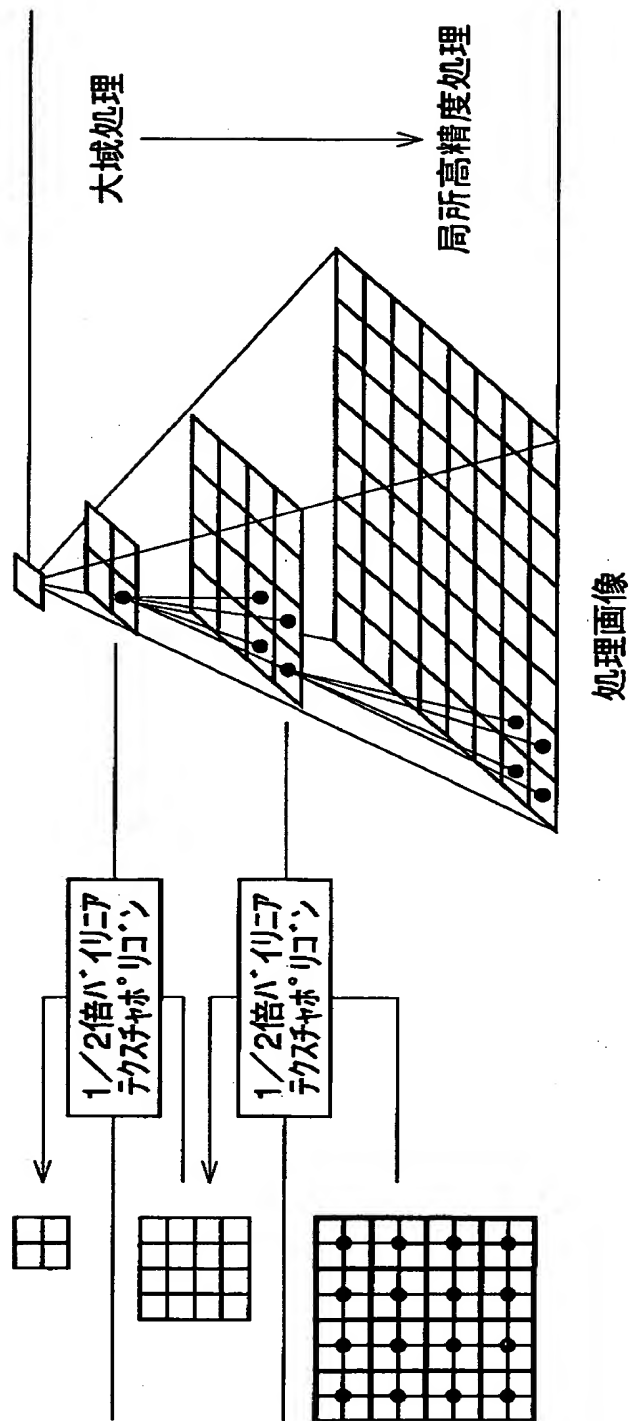
【図16】

		C_{dp}					
		0	1	2	3	4	5 (HMAX-1)
j	0						
	1		$\frac{C_{MAX}}{2}$	$\frac{C_{MAX}}{2}$	$\frac{C_{MAX}}{2}$	$\frac{C_{MAX}}{2}$	$\frac{C_{MAX}}{2}$
	2		$\frac{C_{MAX}}{2}$	$\frac{C_{MAX}}{2}$	$\frac{C_{MAX}}{2}$	$\frac{C_{MAX}}{2}$	$\frac{C_{MAX}}{2}$
	3		$\frac{C_{MAX}}{2}$	$\frac{C_{MAX}}{2}$	$\frac{C_{MAX}}{2}$	$\frac{C_{MAX}}{2}$	$\frac{C_{MAX}}{2}$
	4		$\frac{C_{MAX}}{2}$	$\frac{C_{MAX}}{2}$	$\frac{C_{MAX}}{2}$	$\frac{C_{MAX}}{2}$	$\frac{C_{MAX}}{2}$
	5 (VMAX-1)		$\frac{C_{MAX}}{2}$	$\frac{C_{MAX}}{2}$	$\frac{C_{MAX}}{2}$	$\frac{C_{MAX}}{2}$	$\frac{C_{MAX}}{2}$

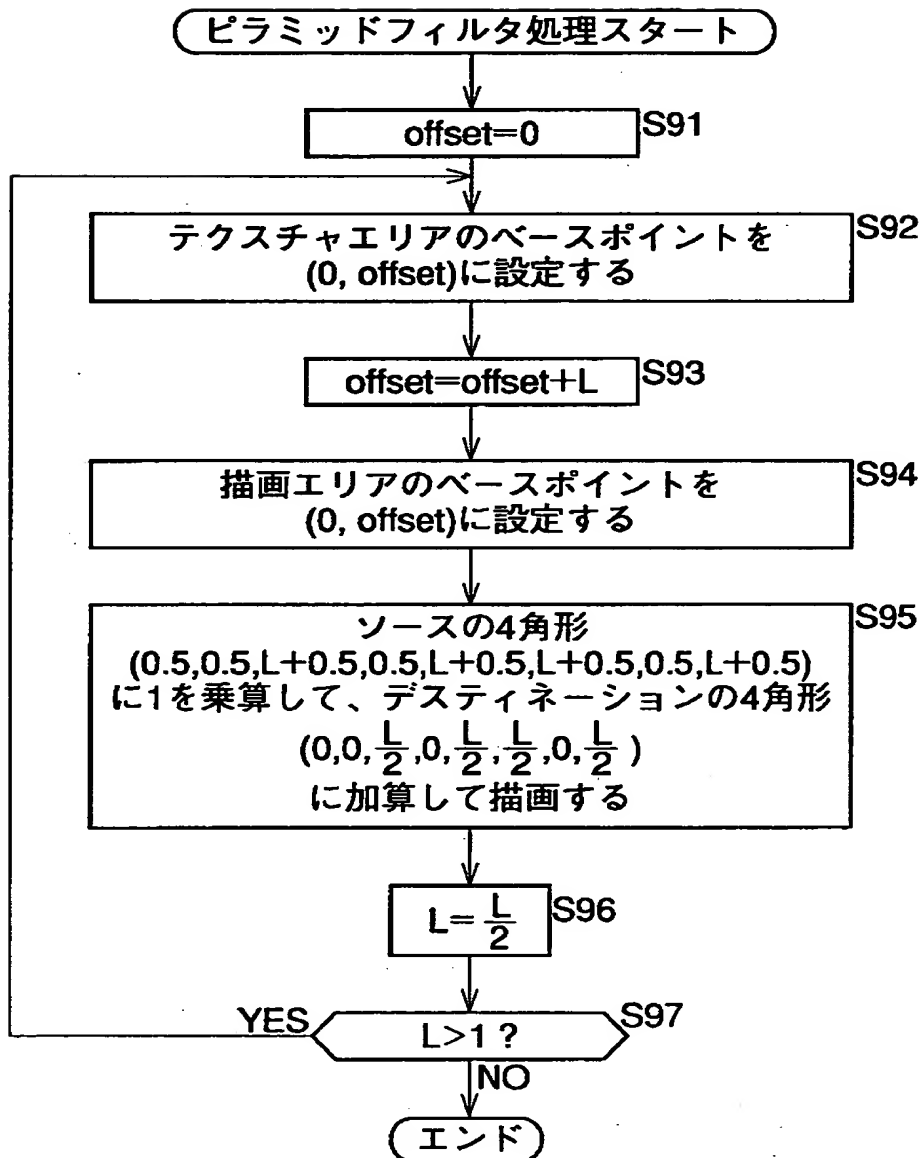
【図17】



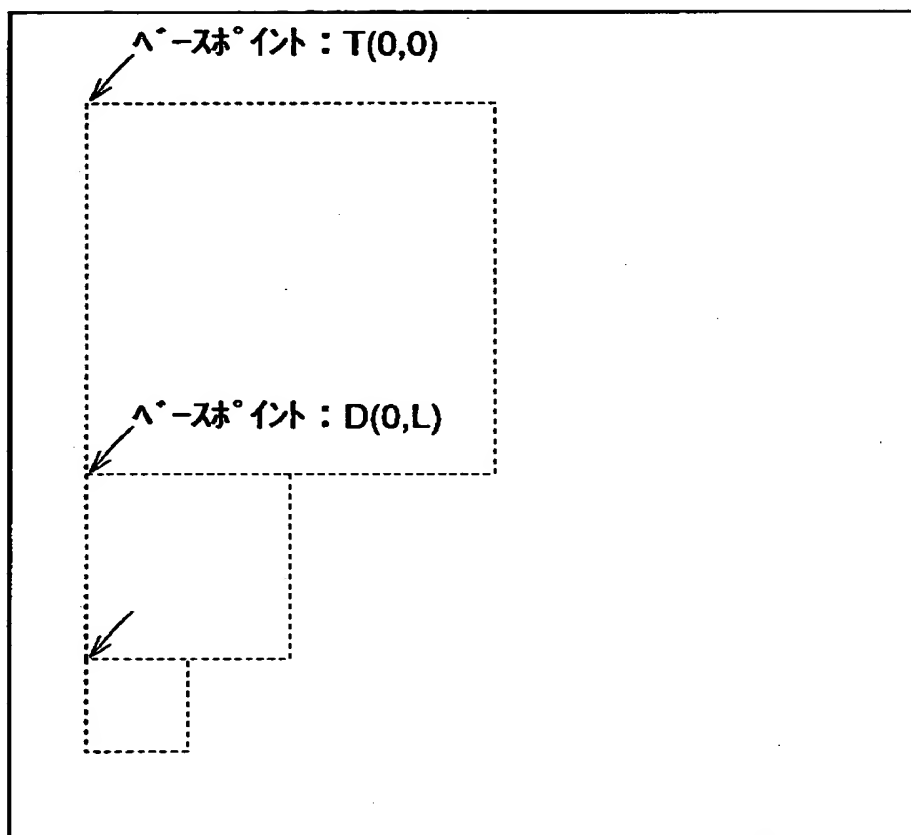
【図18】



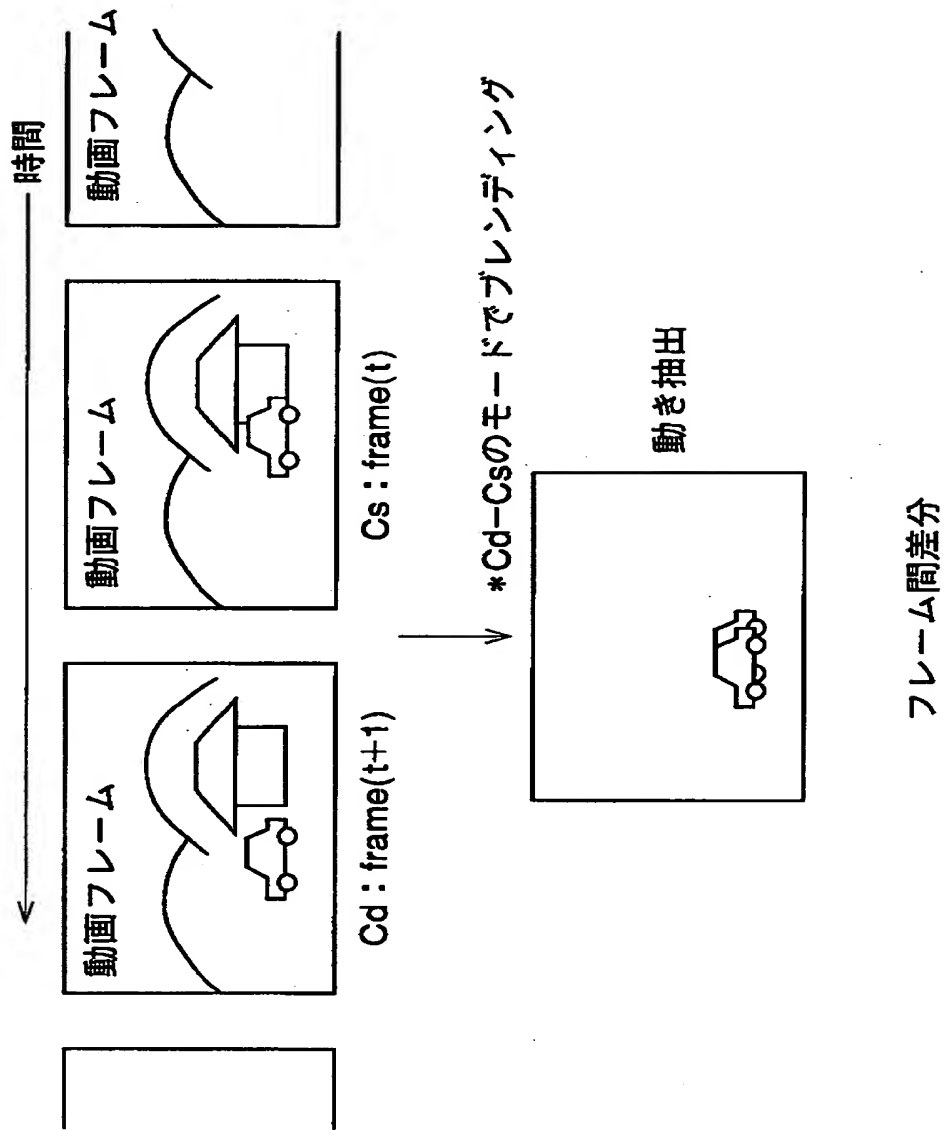
【図19】



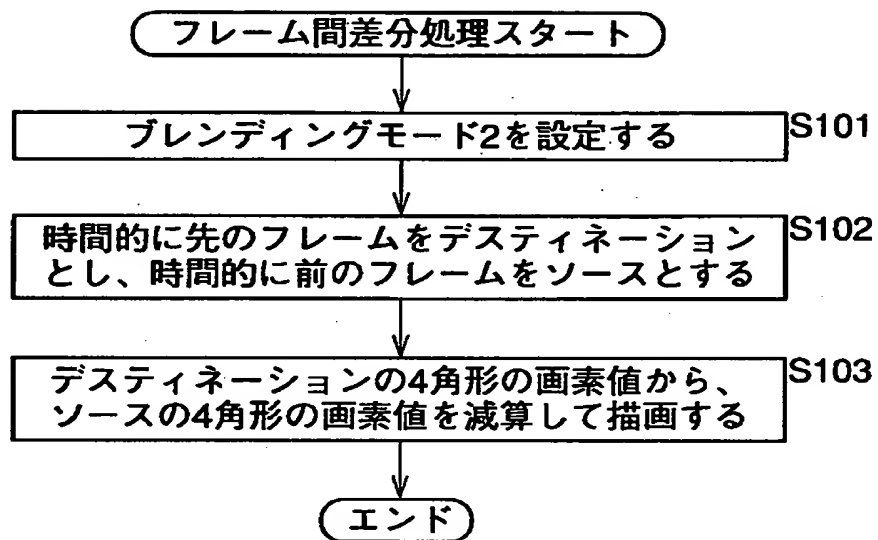
【図20】



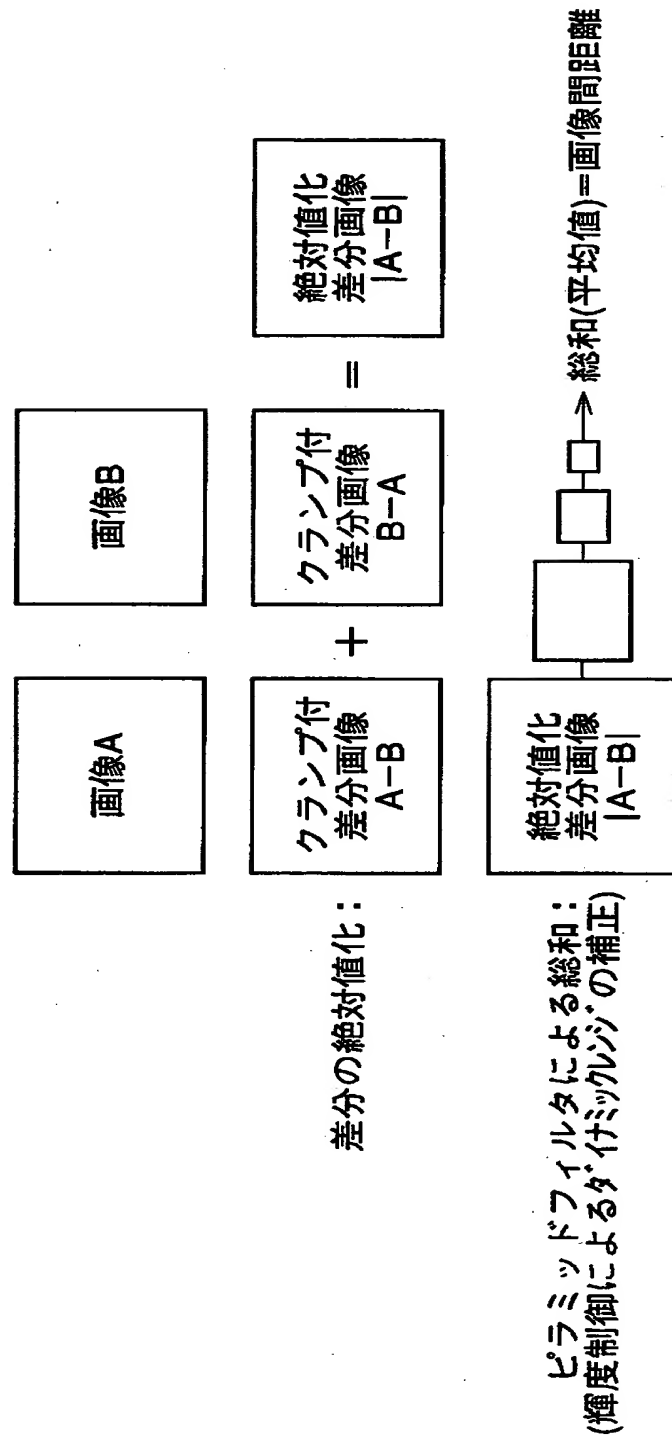
【図21】



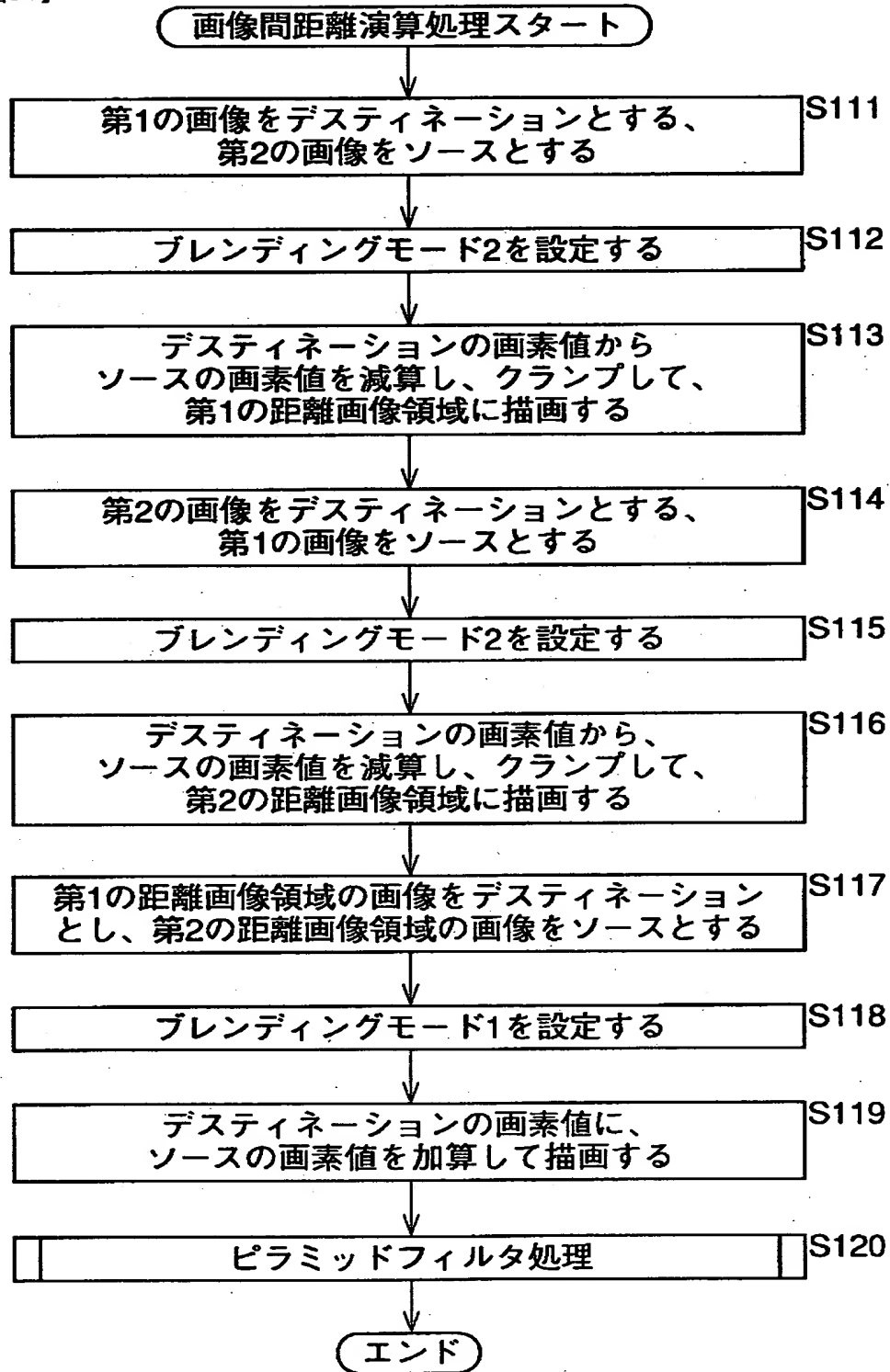
【図22】



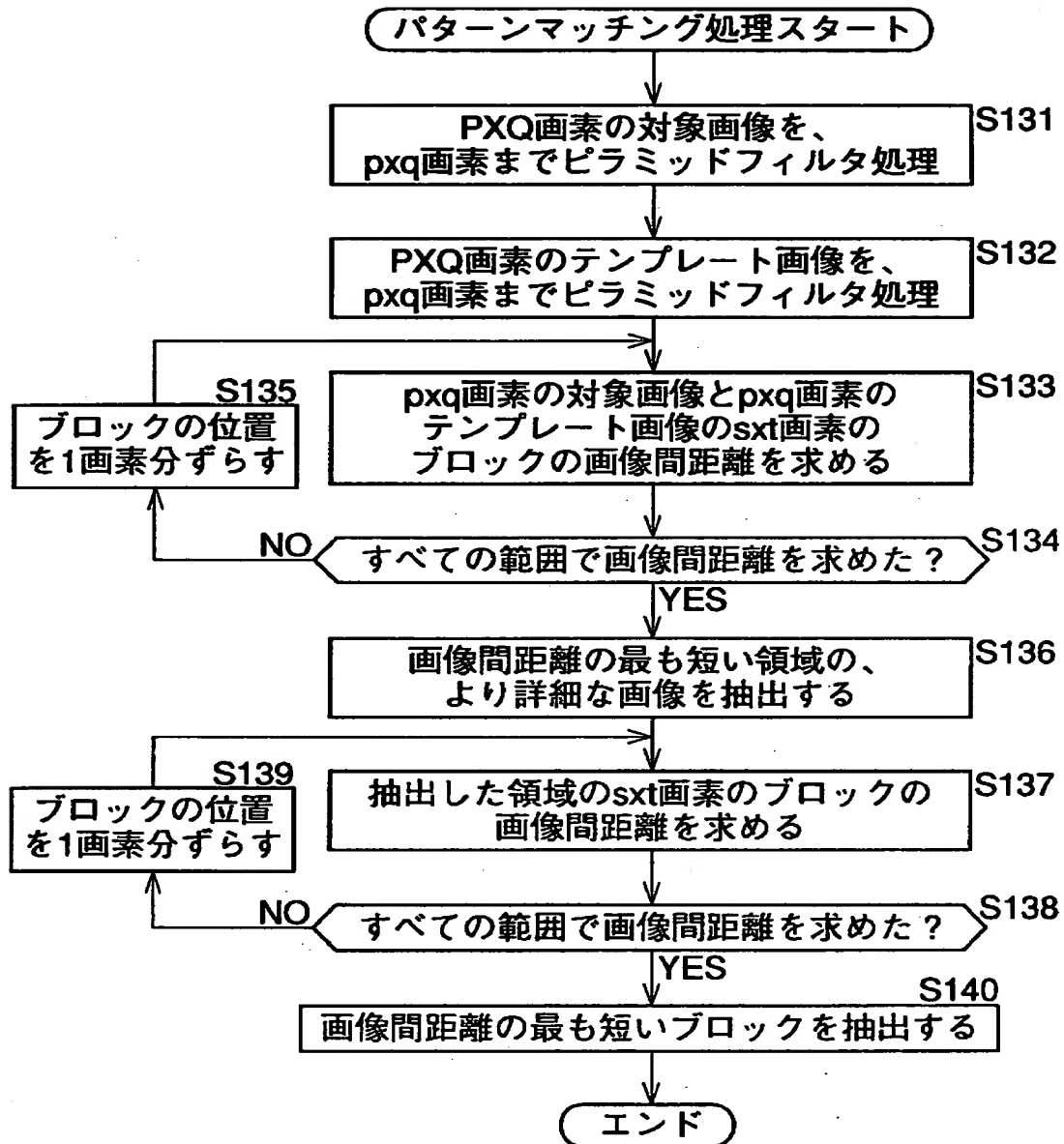
【図23】



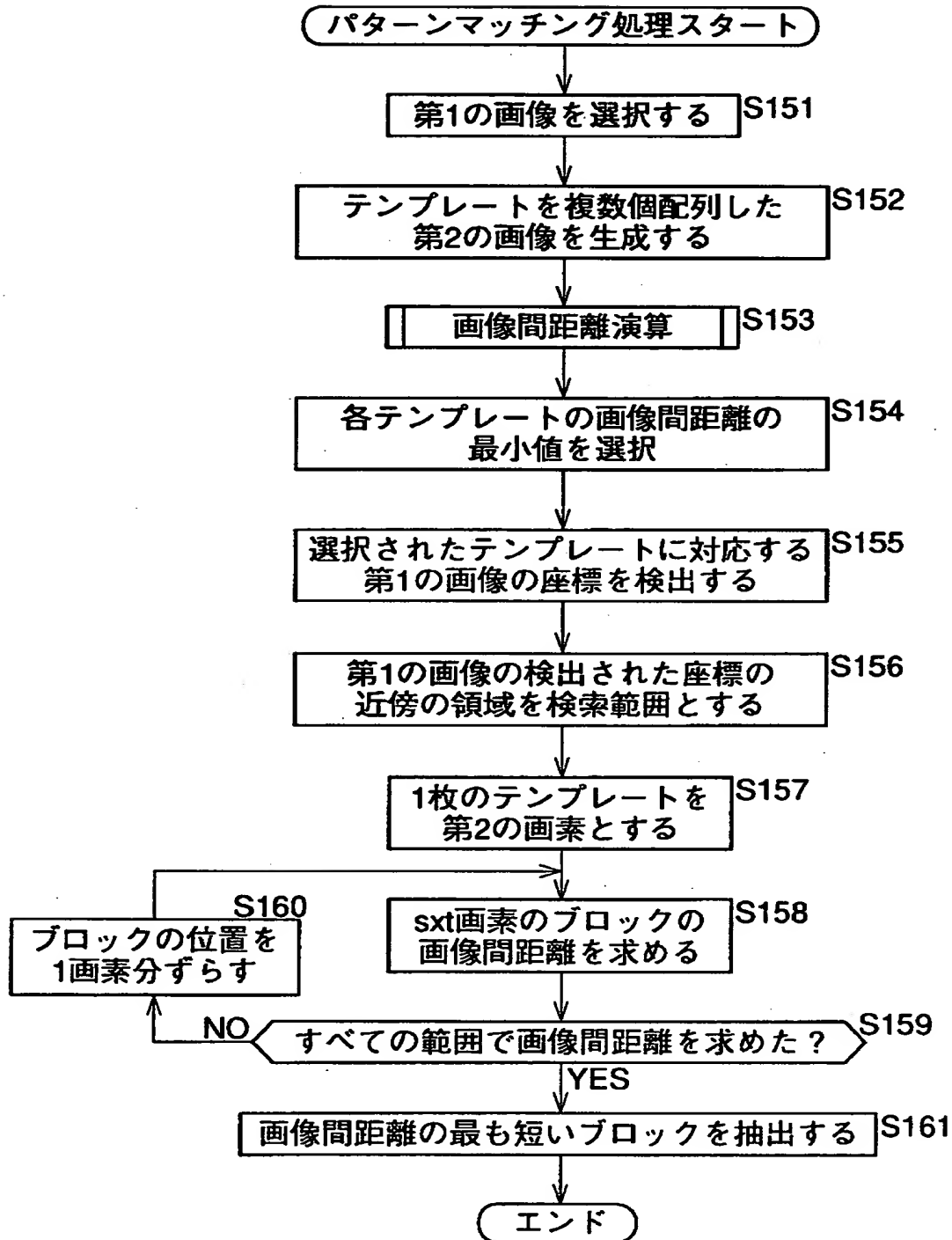
【図24】



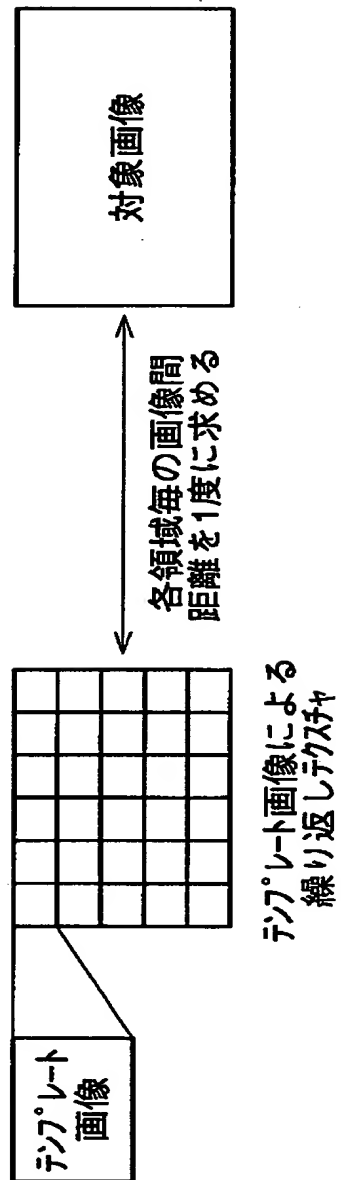
【図25】



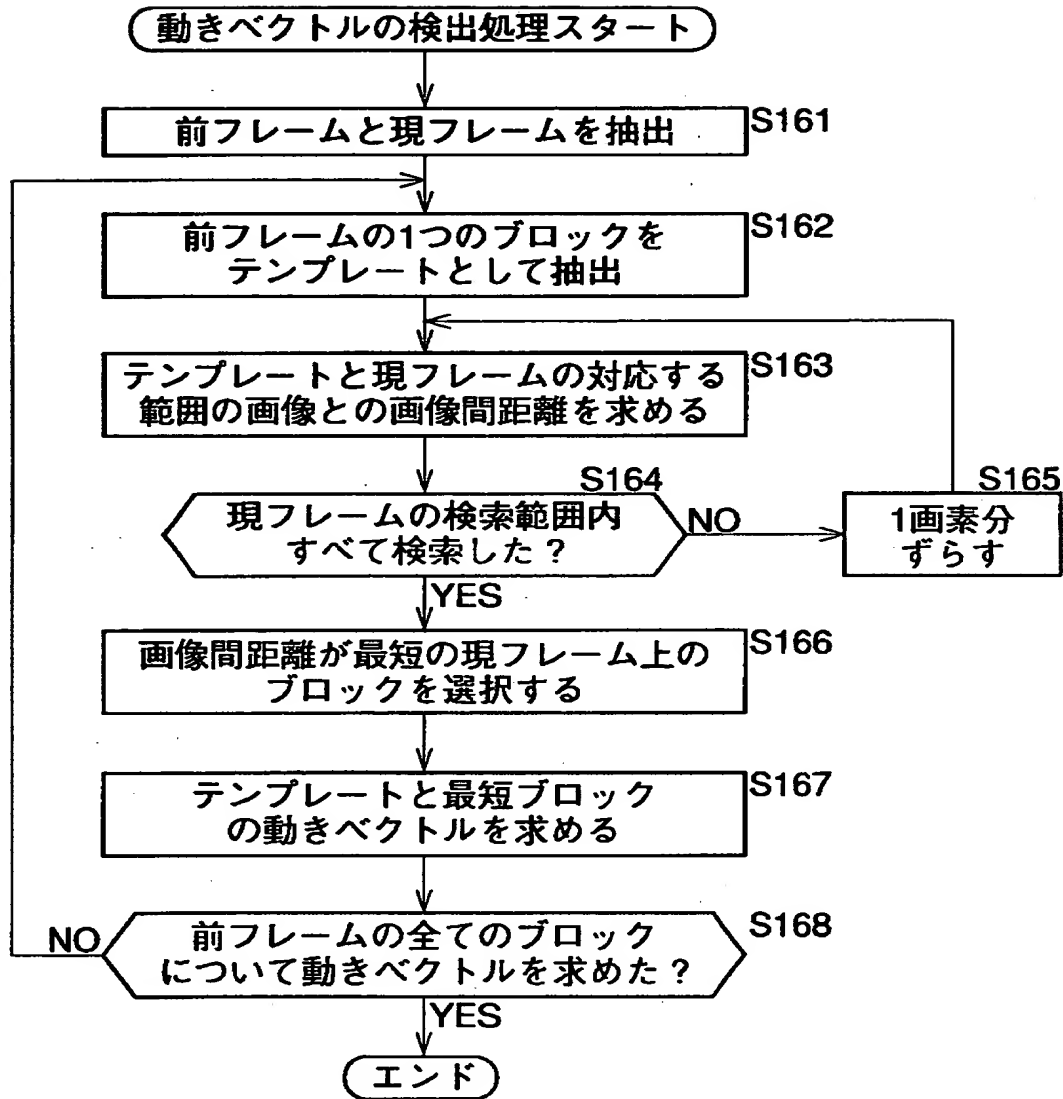
【図26】



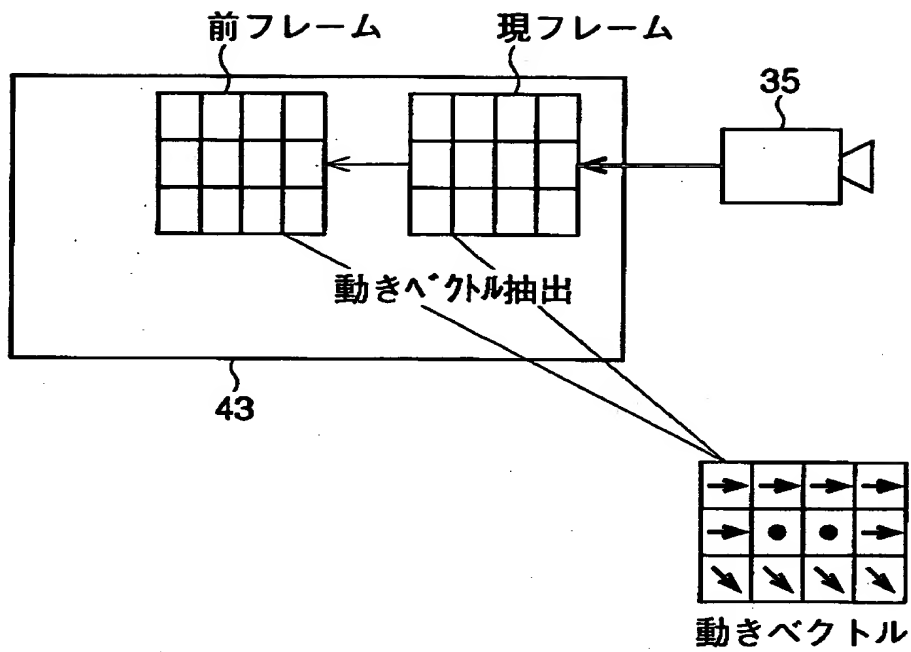
【図27】



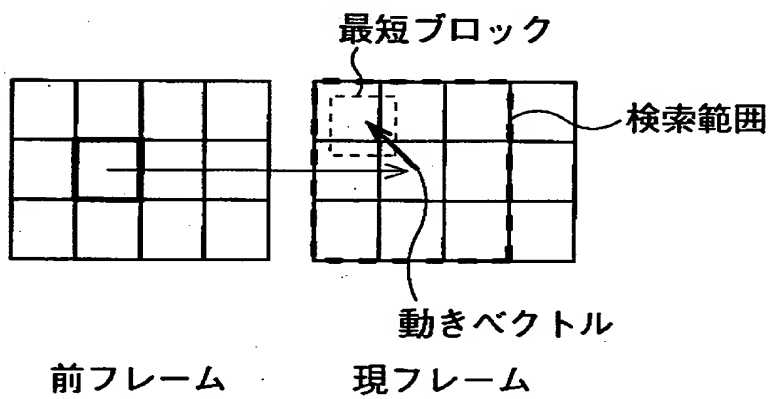
【図28】



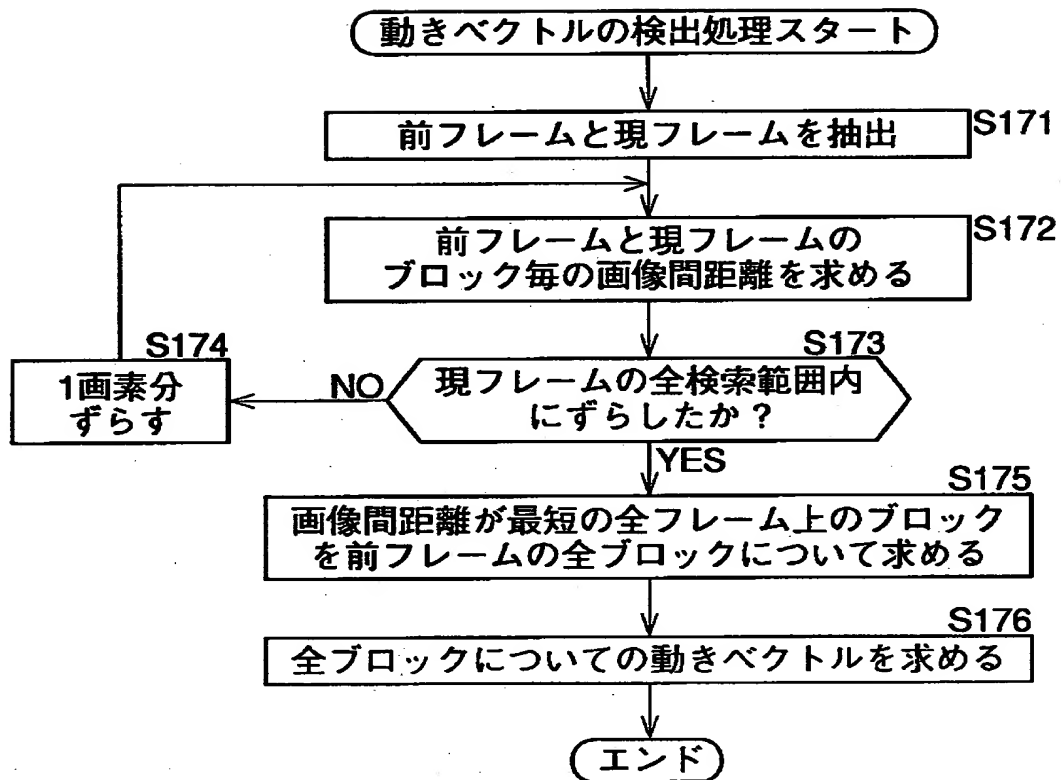
【図29】



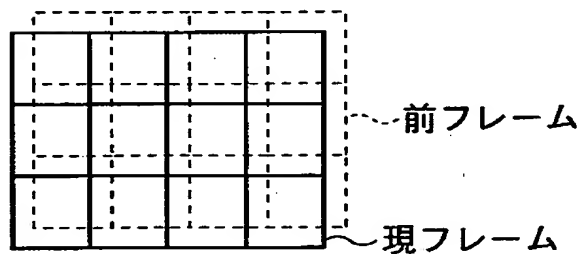
【図30】



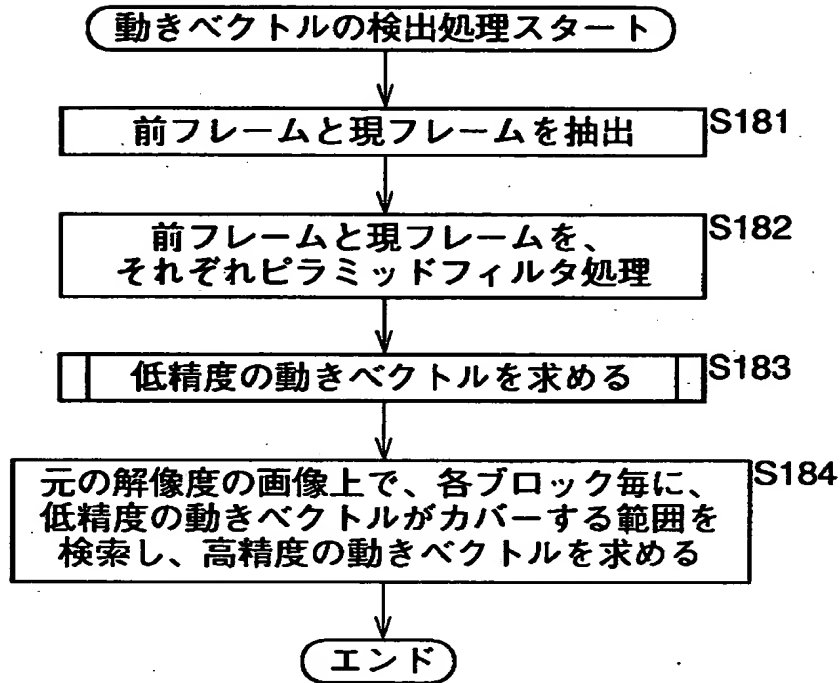
【図31】



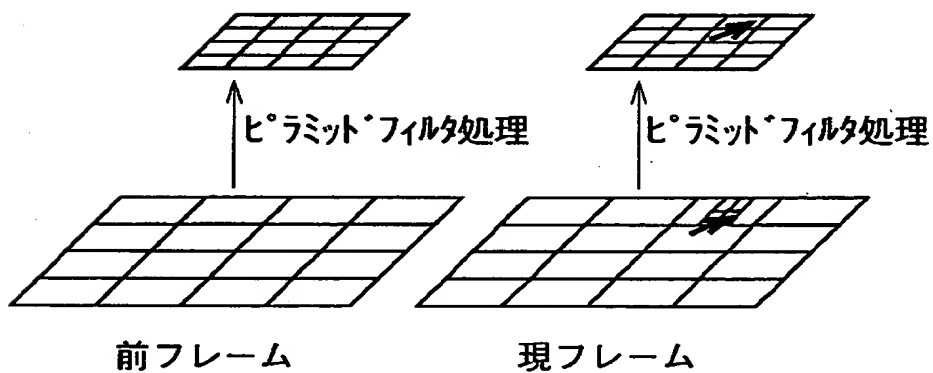
【図32】



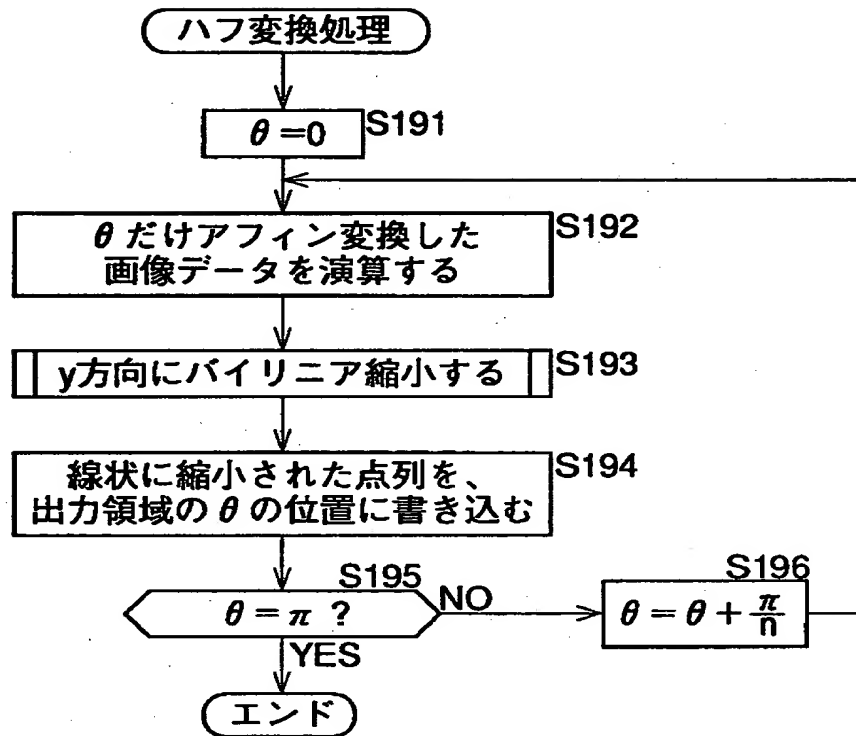
【図33】



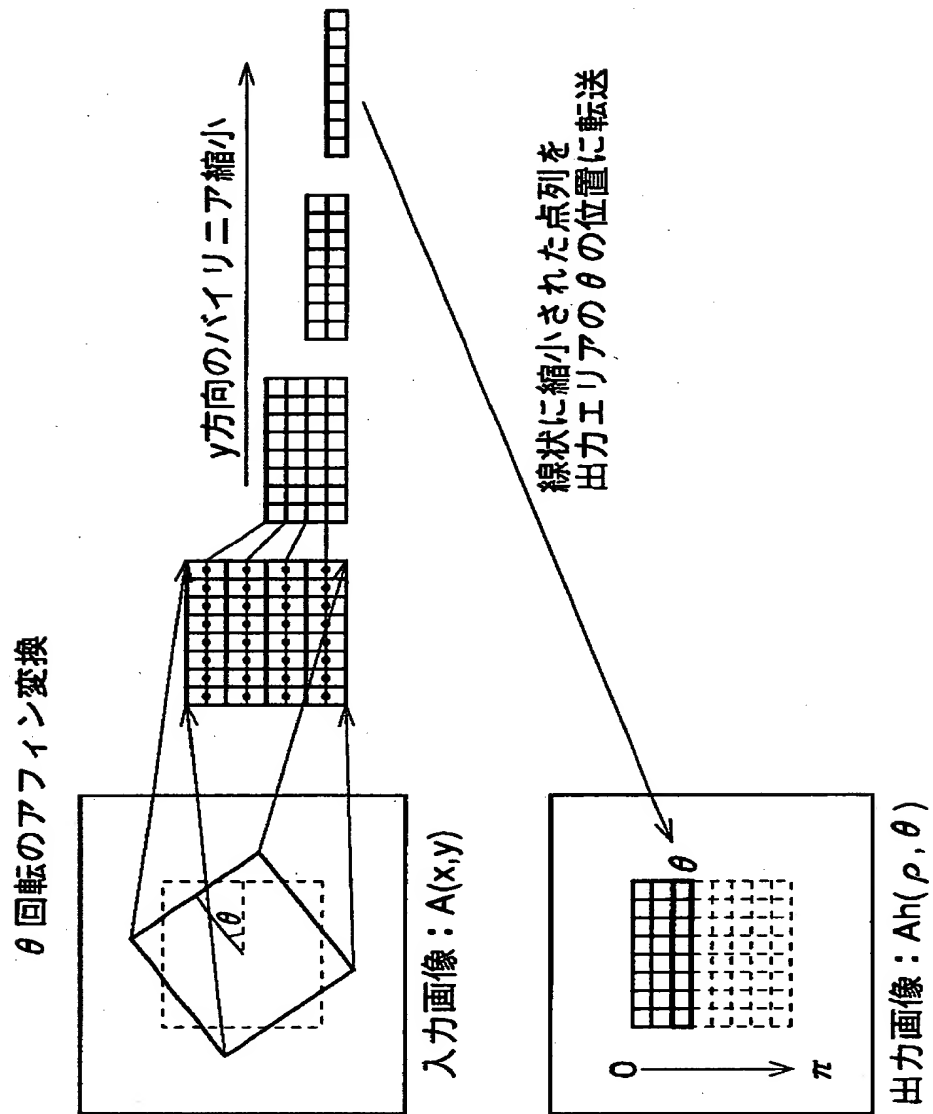
【図34】



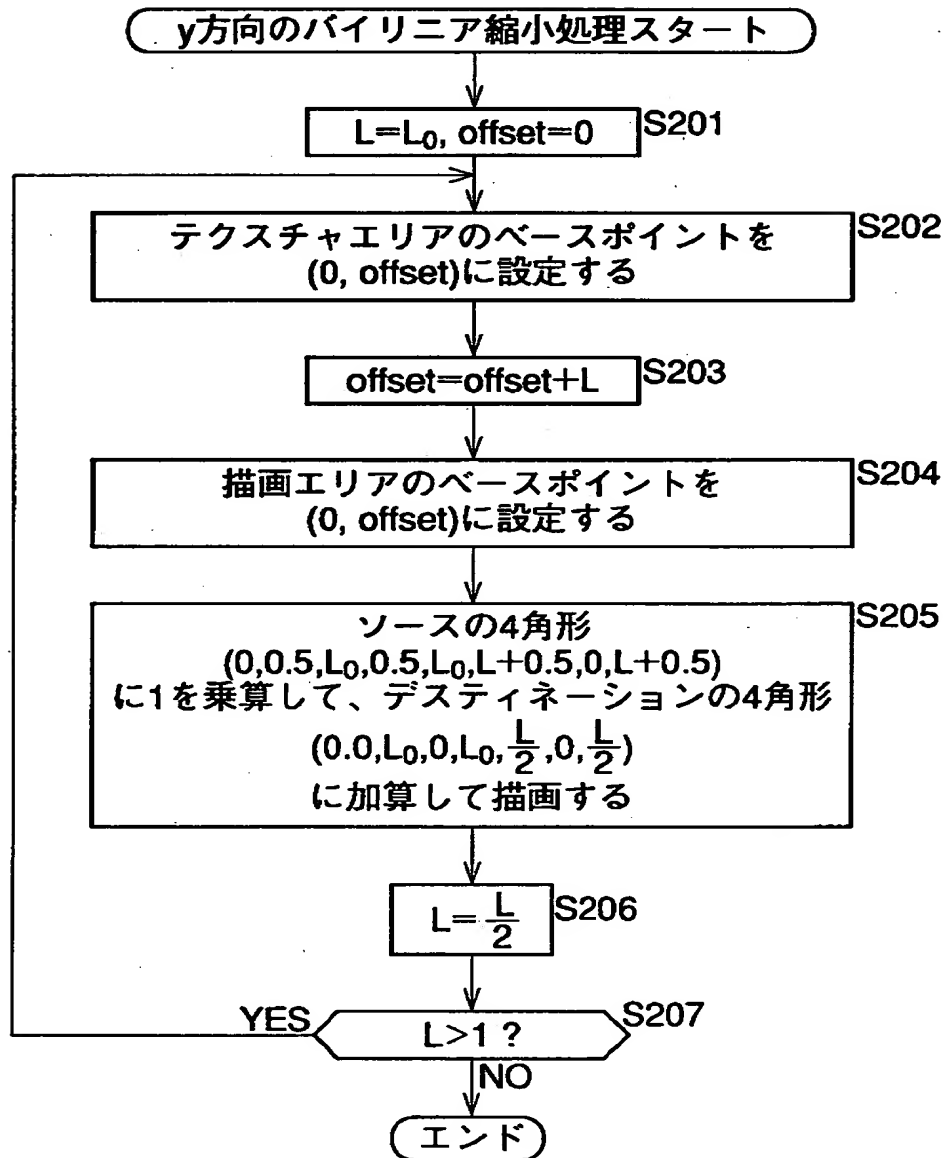
【図35】



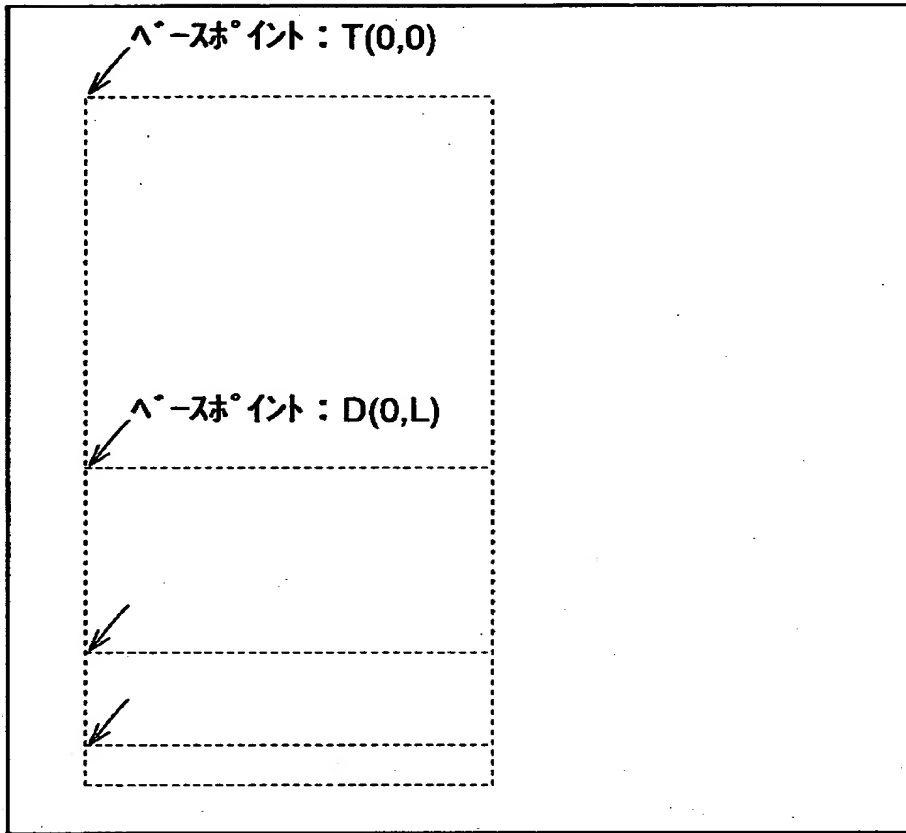
【図36】



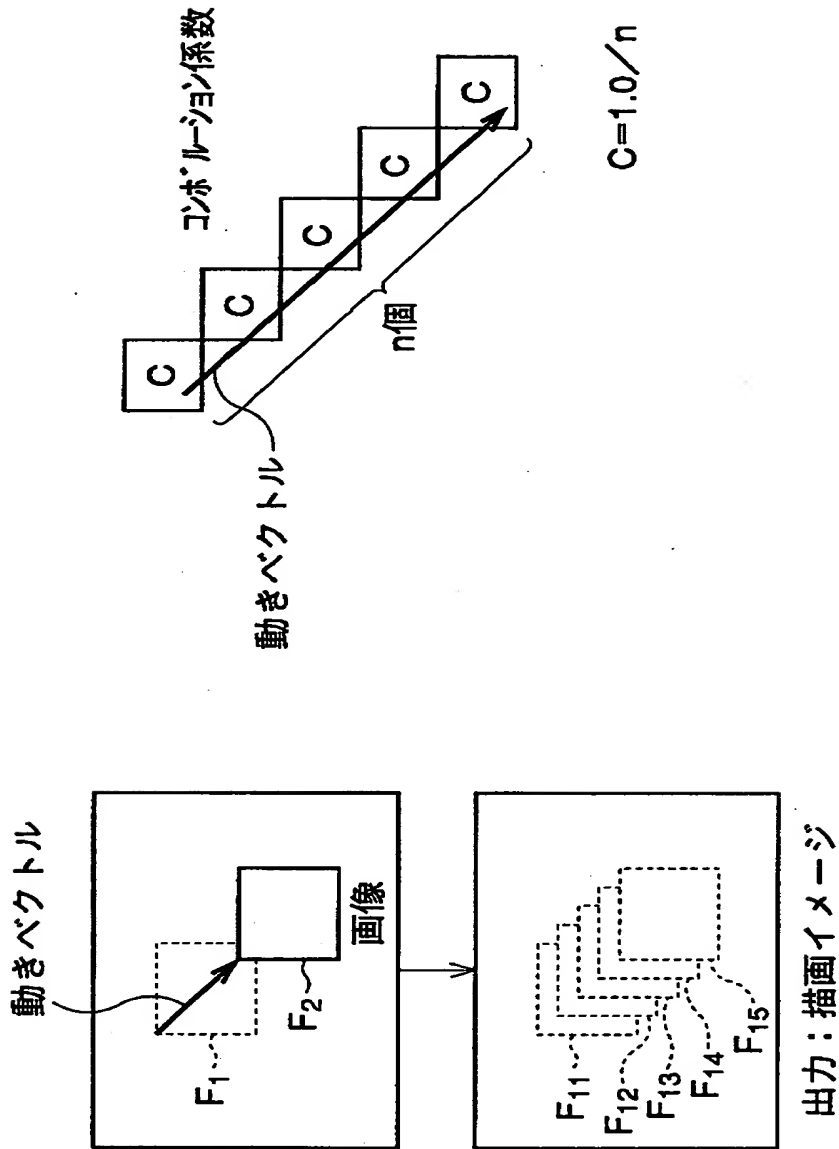
【図37】



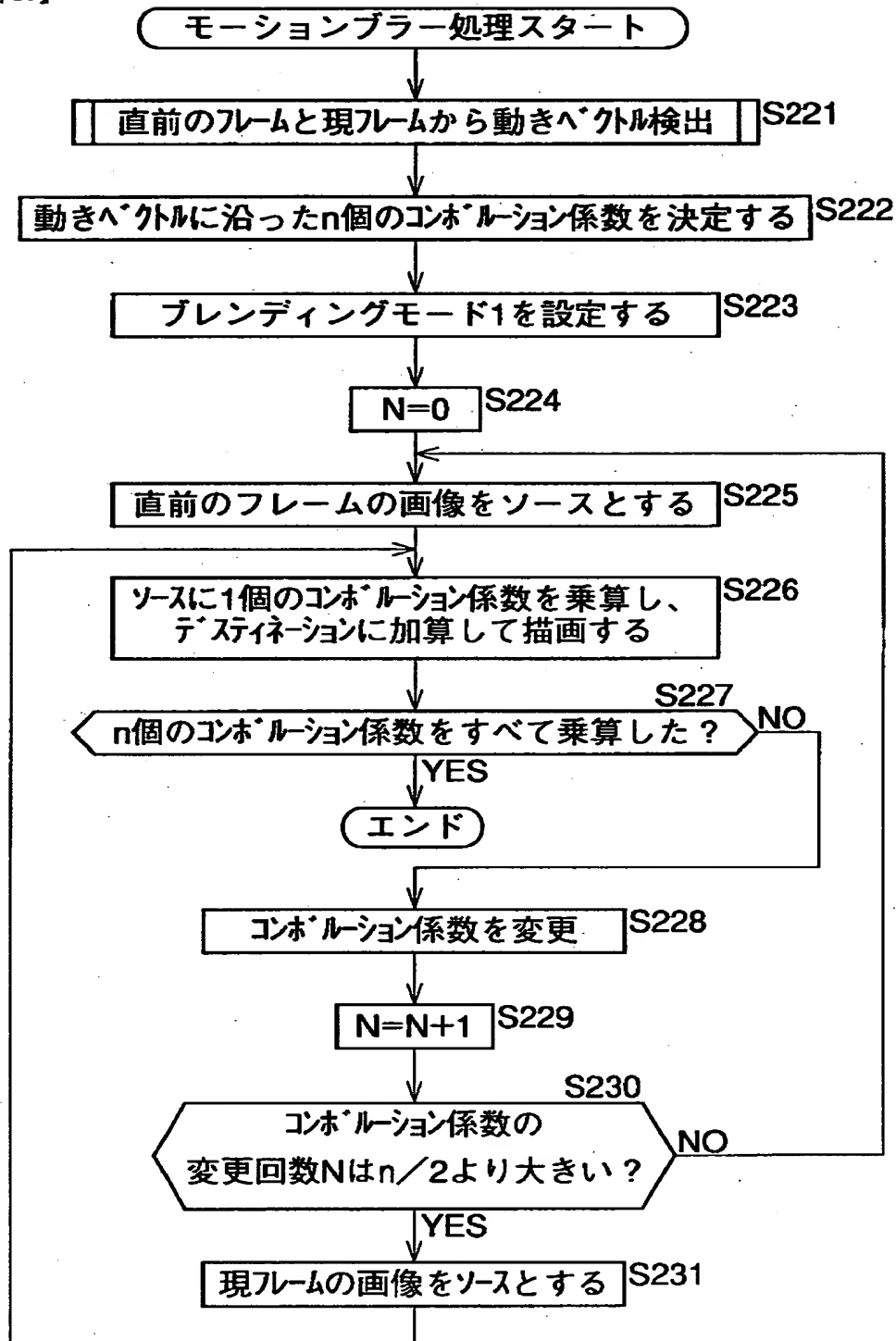
【図38】



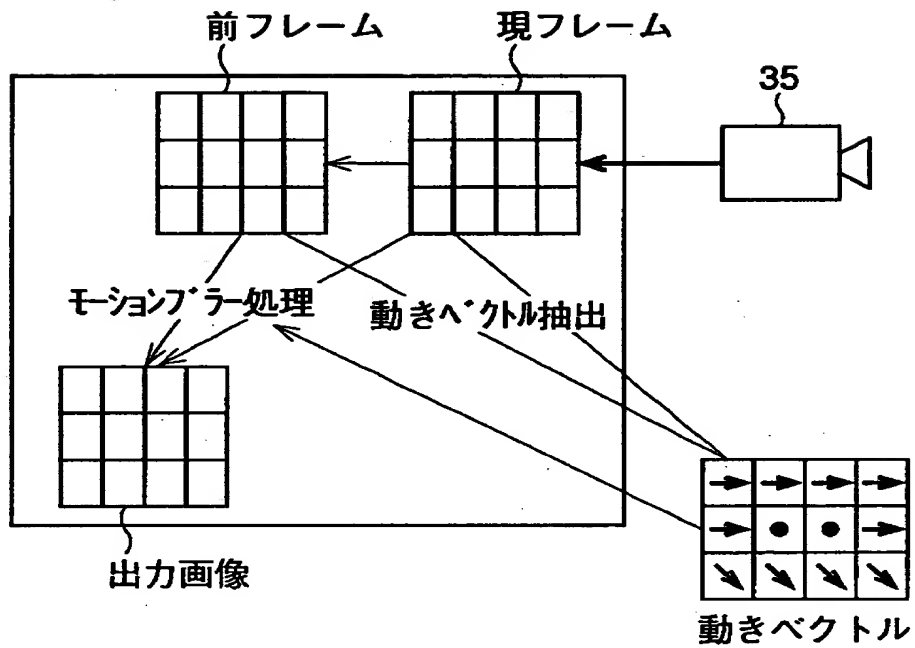
【図39】



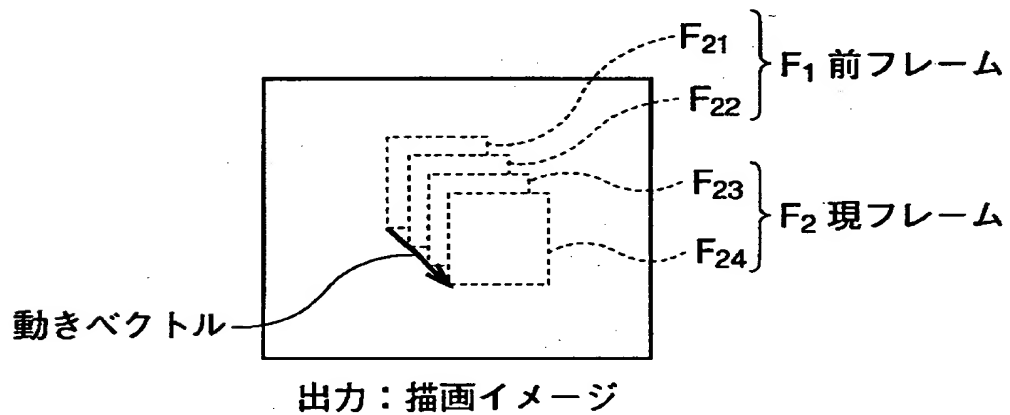
【図40】



【図41】

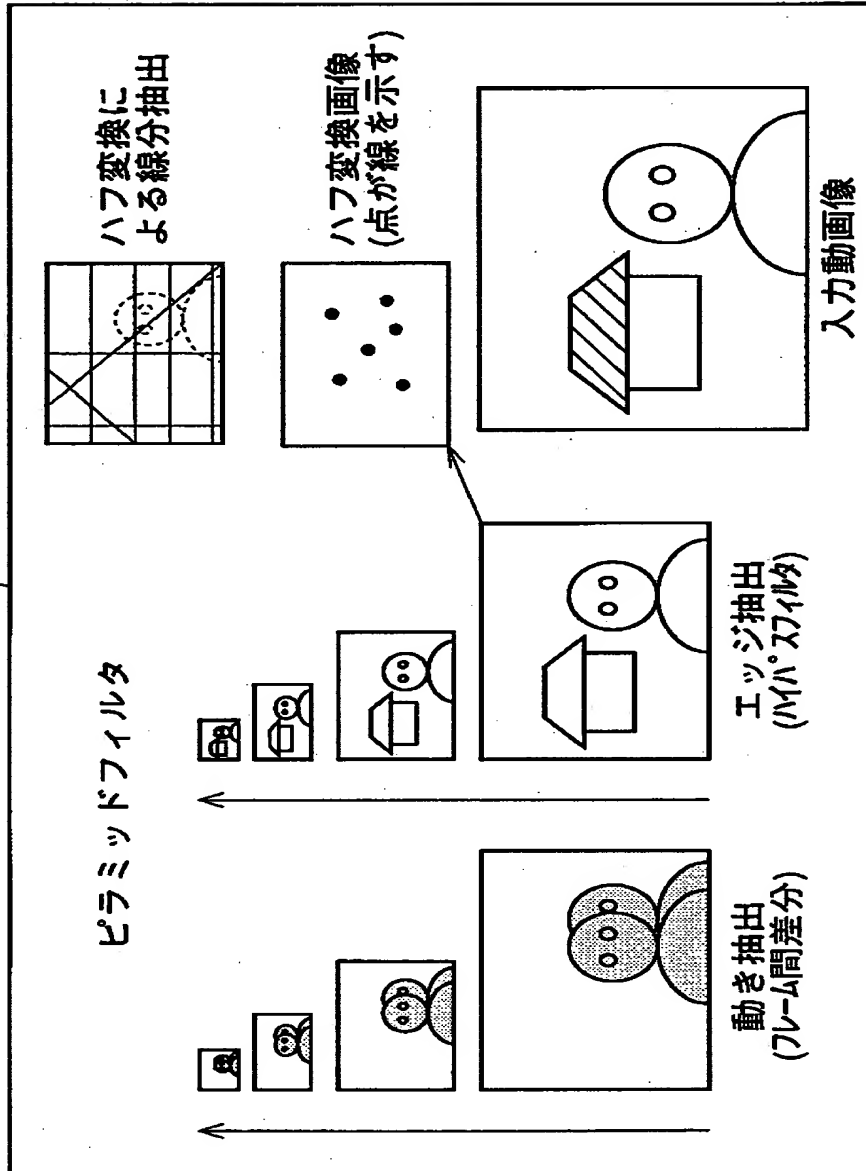


【図42】

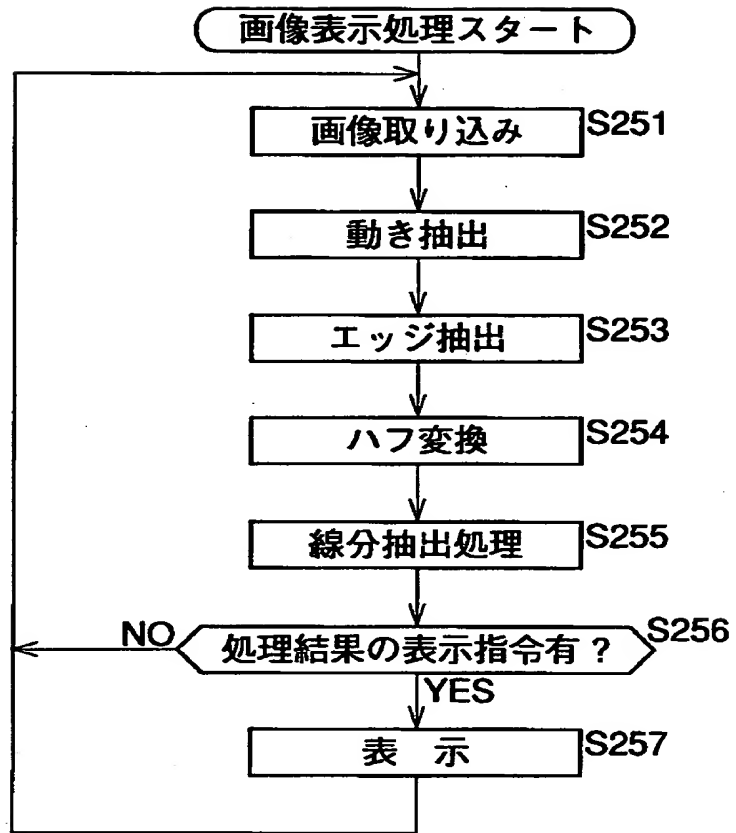


【図43】

36 表示画面



【図44】



【図45】

(A)

0	-1	0
-1	4	-1
0	-1	0

(B)

-1	-1	-1
-1	8	-1
-1	-1	-1

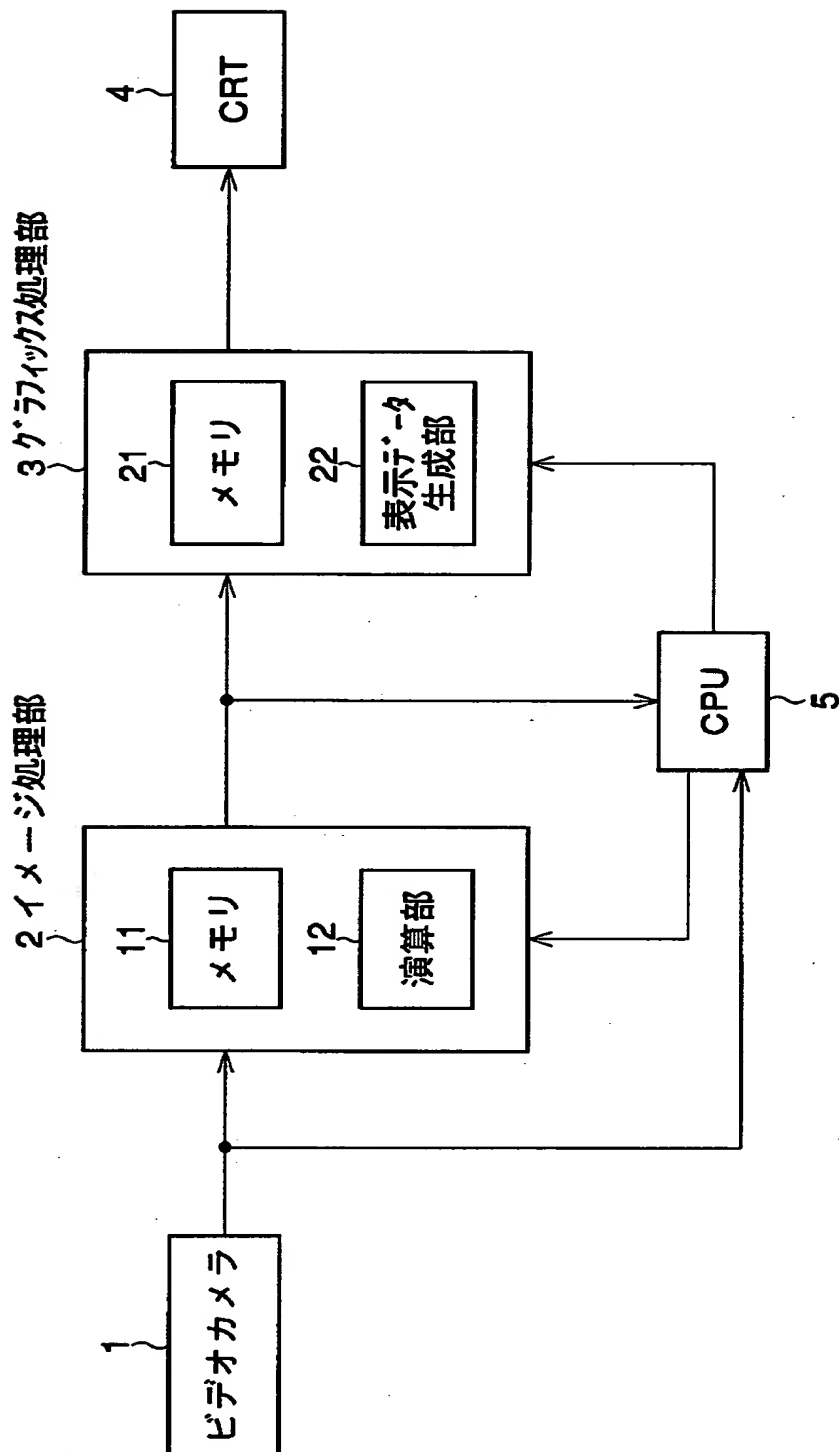
(C)

$-1/4$	$-1/4$	$-1/4$
$-1/4$	2	$-1/4$
$-1/4$	$-1/4$	$-1/4$

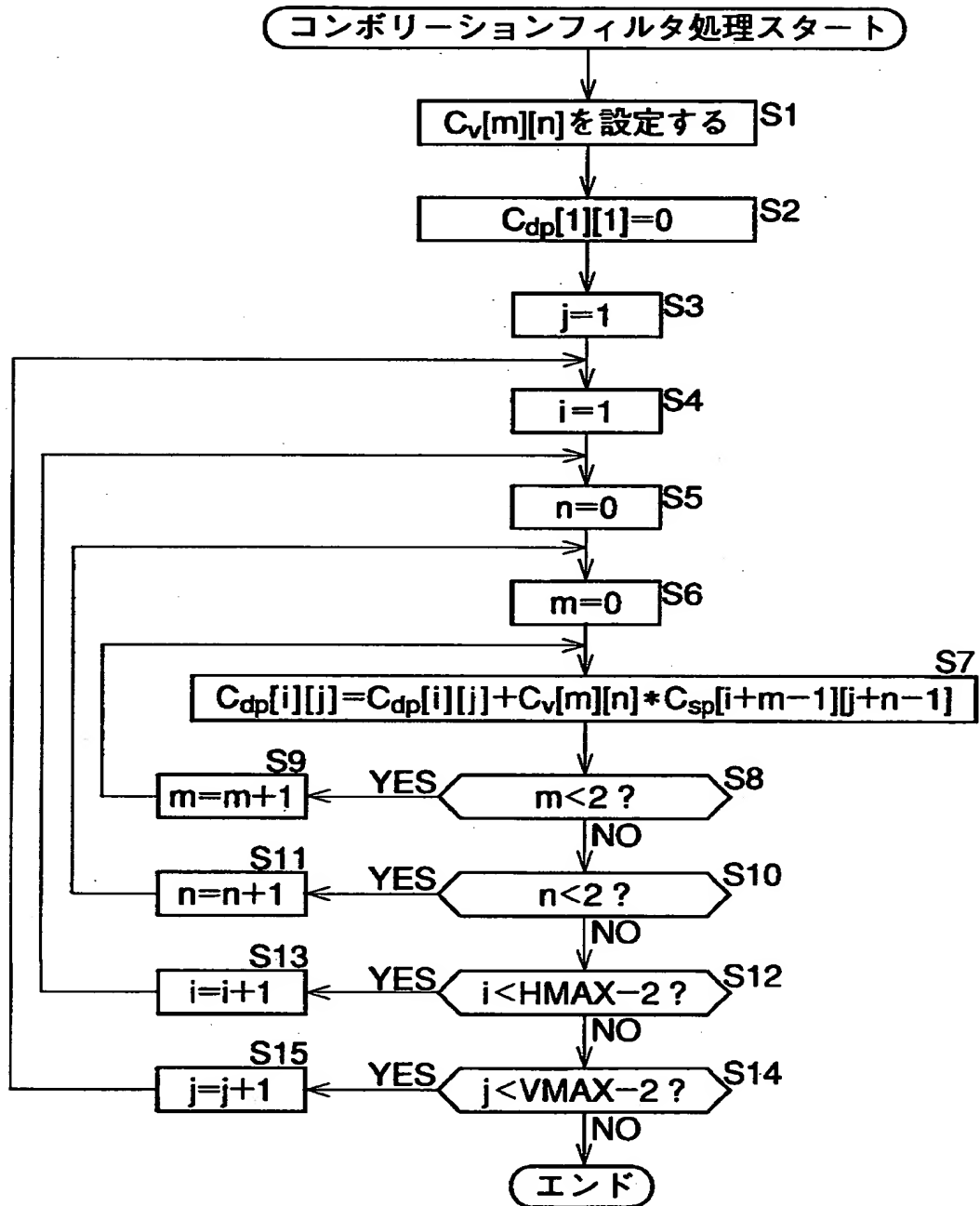
(D)

$-1/8$	$-1/8$	$-1/8$
$-1/8$	1	$-1/8$
$-1/8$	$-1/8$	$-1/8$

【図46】



【図47】



【図48】

		C_{sp}					
		i					
		0	1	2	3	4	HMAX-1
j	0						
	1						
	2						
	3						
	4						
	VMAX-1						

【図49】

		C_v		
		m		
		0	1	2
n	0			
	1			
	2			

【図50】

		C_{dp}					
		i					
		0	1	2	3	4	HMAX-1
0							
1							
2							
3							
4							
VMAX-1							
j							

【書類名】 要約書

【要約】

【課題】 簡単な構成、かつ、低コストの装置で、画像処理ができるようにする。

【解決手段】 メインCPU 31は、レンダリングエンジン41に対して、描画コマンドでコンボリューションフィルタ処理演算を指令する。レンダリングエンジン41は、この指令に対応して、画像メモリ43のテクスチャエリアの画素データにコンボリューションフィルタ係数を乗算した値を、描画エリアに重ね書きする処理を繰り返し実行することで、コンボリューションフィルタ処理を実行する。

【選択図】 図1

【書類名】

職権訂正データ

【訂正書類】

特許願

<認定情報・付加情報>

【特許出願人】

【識別番号】

395015319

【住所又は居所】

東京都港区赤坂 7-1-1

【氏名又は名称】

株式会社ソニー・コンピュータエンタテインメント

【代理人】

申請人

【識別番号】

100082131

【住所又は居所】

東京都新宿区西新宿 7 丁目 5 番 8 号 GOWA 西新

宿ビル 6 F 稲本国際特許事務所

【氏名又は名称】

稲本 義雄

出 願 人 履 歴 情 報

識別番号 [395015319]

1. 変更年月日 1997年 3月31日
[変更理由] 住所変更
住 所 東京都港区赤坂7-1-1
氏 名 株式会社ソニー・コンピュータエンタテインメント